micro

Adventurer

May 1984 75p

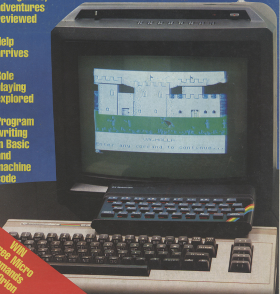
Winning ways for Valhalla

Simulations, war games, adventures reviewed

Help arrives

Role playing explored

Program writing in Basic and machine code



WIN Three Micro Commands from Orion Data

Micro Adventurer

Editor
Graham Cunningham

Assistant Editor
Carmel Anderson

Software Editor
Graham Taylor

Master Adventurers
Tony Bridge
Mike Grace

Editorial Secretary
Cleo Cherry

Advertisement Manager
David Lake

Advertisement Executive
Simon Langston

Administration
Theresa Lacy

Managing Editor
Brendon Gore

Publishing Director
Jenny Ireland

Telephone number
(all departments)
01-437 4343

UK address
Micro Adventurer, 12-13 Little Newport Street, London WC2R 3LD

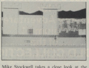
US address
Micro Adventurer, c/o Business Press International, 205 East 42nd Street, New York, NY 10017

Subscriptions
UK £10.00 for 12 issues, overseas surface (excluding US and Canada) £16 for 12 issues, US and Canada air-lifted US$33.95 for 12 issues.

Micro Adventurer is published monthly by Sunshine Books, Scot Press Ltd. Typesetting by In-Step Ltd, 33-41 Dallington Street, London EC1. Printed by Eden Fisher (Southend) Ltd, Southend-on-Sea, Essex. Distributed by SM Distribution, London SW9 (telephone 01-274 8611, telex: 261643). ISSN 0265-4156. Registered at the Post Office as a newspaper.
 Telex: 296275.

SUNSHINE

ADVENTURES ● WAR GAMES ● SIMULATIONS



EDITORIAL

IT'S AMAZING WHO you bump into when you play adventures. For example Humphrey Bogart couldn't have asked for a better script writer than Salamander Software: "My name is Diamond, Dan Diamond. I'm a private cop. I work the Big Apple, a seething metropolis filled with human misery and Chinese takeaways (doubtless even more miserable now that VAT has to be added). Normally I only do routine divorce cases — but when she walked into my office I found myself involved in a case so strange that it made the Big Sleep look like a cat nap." It certainly sounds more interesting than Thorin singing about gold — which he tends to do a lot when we play the Hobbit.

But it's not just who you meet that contributes to the fascination, the plots that the programmers develop for their characters add to the fun. So Salamander's man in a dirty trenchcoat gets to explore a crypt, loses himself in space, and then wakes up space-wrecked on a desert island to indulge in some fishy business. Come to think of it, may be Humphrey Bogart wouldn't have been so keen after all.

And now one of our favourite comic book heroes, Spiderman, is set to make his software debut. So let's hear it for Peter Parker — a hero whose problems centre not on a weakness to Krypton but on a whole bundle-full of adolescent neuroses and a tyrant of an editor for a boss (no, we don't model ourselves on JJ). Spiderman aka Parker was one of the first of an idiosyncratic line of comic book heroes from Marvel — a line which culminated in the wonderful Howard the Duck, another space-wrecked super-hero but with a very low and sardonic opinion of human behaviour.

However, the lure of the traditional adventure is not being diminished by the arrival of web-slingers and trenchcoat-wearers. In this issue David Newton explains how he wrote his Land of Orion, based on the legends of Orion the hunter. As David mentions, at the heart of most adventures is a catalogue of objects to be found and used. Andrew Pepper explains how to make this heart pump more strongly. And if you think you've mastered the skills of Basic programming, take a look at some of Stuart Sampson's machine coding suggestions. But don't be misled by all this programming advice. As the successful adventures (and comic book heroes) prove, and as Laurence Miller points out, it's the creative spark that matters. The best of programming is to no avail if you can't capture the user's imagination. And this is not simply a question of adding graphics to a text adventure. The Hobbit comes without pictures on the BBC, but this may be no loss at all. The information necessary to solve the game is in the text, you're free to visualise the characters according to your own view, and the game moves faster.

LETTERS

Send your hints, successes, complaints and compliments to Letters Page, Micro Adventurer, 12-13 Little Newport St, London WC2R 3LD

Quill for children ...

MR FIDDLER (Letters, March 1984) and your other readers may be interested to know that the company Scientific Software is working on an alterable adventure aimed at the younger child. We are still working on the graphic adventure, so it will be a while before the alterable one becomes available, but we think it will be well worth the wait.

K J Fenning,
Scientific Software,
Old Loom House,
Back Church Lane,
London E1 1LS.

has timely arrival

IN REPLY to Mr Fiddler (Letters, March 1984) who asks if there is an adventure program designed for young children, tested in classrooms, that can be used like The Quill and enabling children to write adventures, the answer is yes.

I have been developing such a program over the past two years.

It's called The Tombs of Arkenstone. It was designed as a computer-based classroom and home education activity. It includes a storybook that sets the scene and gives clues to the adventurer.

The Tombs of Arkenstone invites children into a world of dragons and hobgoblins. Included in the package is Make Your Own Adventure. It enables children to write their own adventures and play them.

The concept was demonstrated at last year's MAPE (Micros and Primary Education) national conference, and it was the enthusiasm of the teachers there that motivated its publication. The publishers are Arnold_Wheaton Software, Leeds.

Bab Hart,
The Pines School,
Divot Place,
Hertford.

Making contact

WITH reference to David Swain's letter (February 1984) about the Adventure Contact column: I wrote asking for help with Adventure 200 just before Christmas. The letter was published in the column and I received seven letters in reply. Four in varying degrees were able to offer help, and asked for help in return. Three realised I had progressed further than they and also asked for help.

I answered each letter, although only one writer enclosed a SAE, but as a result of give and take I have solved Adventure 200.

I agree with Mr Swain that it is much easier to communicate by phone. But it also seems reasonable that one should include an SAE and write for help. I gave my phone number to the people who wrote to me.

If anyone needs help with Adventure 200 I shall be pleased to answer any letters.

Irene Feeney,
Surrey Way,
Laindon West,
Essex.



Games writing

I READ the article A beginner's guide to games writing (March 1984) with interest. I own a BBC model B and have found the following method useful for adventure programming.

I use a two dimensional array POS$(X,Y). Before inputting your command the computer reads all the data so no restoring needs to be done while the main program is in action.

The X co-ordinate is for north-south movement and the Y co-ordinate is for east-west movement. It seems much quicker, but do other readers know of any snags? Does anyone else use this method?

Anthony Foss,
10 Ellesmere Rd,
Twickenham,
Middlesex.

Valhalla pointers

I WONDER if your reviewer Graham Taylor or Legend could clear up a couple of points about Valhalla.

When I originally bought the game I spent some time mapping the location and searching for Ofnir and the magic key, but without success. I did, however, find Felstrong and this led me to believe that while the objects must be found in order, it was not necessarily in the order they were given in the manual. However, Mr Taylor's review suggests that they should be in the manual order.

Can anyone clarify? Can the objects be found in any order? What use is Felstrong? I have found many places needing Ofnir or Grimnir to enter, but none where I can use Felstrong.

John Davidson,
Bretton Drive,
Broughton,
Chester.

SEE Adventure Help last month for a few clues to the gaining of Ofnir, the first quest object. Many people seem to find Felstrong first, contrary to the manual. Legend assure us that this is the only case of the order getting out of kilter.

As far as I know, Felstrong is no importance.

Prisoner theory

I READ with great interest the two articles on *The Prisoner* by Mike Grace. My friends and I have analysed and re-analysed the series and I have come with the following invaluable information:

Number 1 is either Terry Wogan, Margaret Thatcher or Sir Clive (Sinclair).

The Butler is Lena Zavaroni in disguise.

The Village most likely is Milton Keynes

The Prisoner is Lord Lucan. (Now we know where he's gone).

Timothy Clyne,
Fraserburgh,
Scotland.

Thoughtful Quest

FIRST OF all congratulations on producing a magazine that fills a gap in the computer magazine range.

You asked readers to write in about favourite adventures. I would like to mention an adventure from Phipps Associates for the Spectrum 48K, Knights Quest, which is every bit as puzzling as The Hobbit.

Steve Jones,
Highland Rd,
Bromley Cross,
Lancs.

NEWS DESK

If you think you've something newsworthy, call 01-437 4343 and let us know

UK loss is US gain

THE HIGH cost of telephone calls is the only obstacle preventing UK adventurers from playing a real time, interactive game.

Known as the Multi-User Dungeon (MUD) it has been running at Essex University for the past five years. It was written in 1979 by a computer science undergraduate, Roy Trubshaw, but most of the time it has been operated by post graduate student Richard Bartle.

MUD was written for and is played on a DEC System 10 mainframe computer, which the players use on a time-sharing basis.

Richard said there was really no limit to the number of players who could play the game, but that the numbers obviously had to be kept to a rational level.

Having worked on the game for the past four years the next stage in its development according to Richard, is to make it available.

"But telephone costs are too expensive here for people to play a real time, interactive game," he said.

"The Multi-User Dungeon is to be marketed on large networks in America. Costs are low enough for people to use the phone for two to three hours at a time."

In 12 to 18 months Richard will try to market the game in other countries, but it looks as though telephone charges may continue to exclude UK players from Multi-User Dungeon.

Bringing games to the office

AN AMERICAN firm has come up with what appears to be the ideal adventure for the office.

Called Rogue, it coems on disk and runs on the IBM PC. You move your character around different levels of mazes, avoiding trolls and quaggas, in an attempt to find the Amulet of Yendor quickly.

The game is reported to have a wide range of commands — including S (for Supervisor).

Press this command and the dungeon map disappears. It's replaced by a dummy of the PC-DOS operating system to fool prying eyes — but don't sue us if you get caught out in the office.

Rogue comes from American software house Artificial Intelligence Design Systems. We've no news on UK availability, but if IBM sells more PCs here, then perhaps the adventure will follow.

Aid for Hobbit fans

IN A HUMANE gesture to Hobbit fans, who may have lost their sanity in the goblins' dungeon or over Gollum's riddle, Melbourne House have published a book titled *A Guide to Playing The Hobbit*.

Author David Elkan wrote the book as a guide for both the beginner and the advanced adventurer so that they can improve their understanding of the game.

The book is divided into three sections. The first details the general strategies and tactics needed to play the game. The second and third sections give solutions to problems that are likely to be encountered.

Melbourne House stress that the book will not detract from the challenge of the adventure.

"It does not supply the solution to The Hobbit. It only offers one of the many possibilities," a company spokesman said.

A Guide to Playing The Hobbit is available from most book shops now for £3.95.

Making dreams come true

IF YOU find bar room brawls, fighting monsters, combat using metal weapons, manhunts, archery and primitive living conditions appealing read on.

Set in the Cheshire countryside is the medieval Peckforton Castle where you can brave real life adventures for a few hours or a few weeks.

Organised by Treasure Trap, you can enter the 13th century for £46 — the cost of club membership. As well as an adventure this entitles you to all the facilities that were available to peasants in the 13th century — which were few.

But a Treasure Trap organiser said one concession was made to the 20th century: a shower.

The Basic dungeon at Peckforton Castle gives prospective members a taste of the club's activities. You are equipped with a weapon, a shield, a spell and a miracle. These should give you an idea of the abilities of the four main character classes.

The fee is £10 a person. On completion of the adventure it is deducted from the membership fee.

After completing the adventure the participant chooses a persona and a name for his or her future character, and embarks on a new quest.

Adventures can span 16, 24 or 48 hours. There are weekends set in the wilderness anywhere in Britain and fortnight-long holidays where the hardy can participate in the intrigue of king and court, peers and knights and guilds and clans.

Anyone with £46 to spare and a taste for things medieval can write to Treasure Trap, PO Box 130, Chester.



Machine code for novices

DUNGEON Builder from Dream Software Ltd allows a computer novice to write graphic adventures using machine code.

Taking two years to develop from conception to completion, Dungeon Builder complements the text-writing function of The Quill.

A grid visually aids the writer then placing objects in their locations and enables him or her to see where the player is when writing the game.

The reverse side of the cassette is a utility program for making stand alone games which can be run on a Spectrum without the Dungeon Builder.

Dungeon Builder is accompanied with a 100-page illustrated manual. The package costs £9.95 and is designed for the Spectrum 48K.

According to the Dream Software Ltd managing director Leslie Graddon, users can sell any of the programs written using the Dungeon Builder, but are asked to credit the company on the title page.

Programs to retail at £1.75

EIGHTH DAY Software are producing a series of six adventures which they will sell in summer for £1.75.

The games are being written using The Quill and will range in difficulty from the beginner to the advanced stage.

A director for Eighth day Software, Gary Kelbrick, said the company could afford the cheaper rates by selling the games by mail order and perhaps through newsagents.

The series begins with a game of moderate difficulty called Cuddles. The player is a baby who escapes with its one-armed teddy bear from the nursery.

The aim of the game is to find the teddy bear's arm, return safely to the nursery and to the arms of a distraught nanny.

The second game is a space adventure called Quann Tulla, after the spaceship in which you travel. The player is the only living crew member left on board the craft, which is about to be overtaken by marauding aliens. You must program the ship to explode before escaping on a space hopper.

The third game is a spy story, still in the planning stages. The fourth is a treasure hunt and the "hardest game" in the series, according to Gary. Called Faeriey it is set in Middle Earth.

The fifth game is still being written. It is based on a Stephen King book called *The Stand*, which is concerned with biological warfare. The sixth game, as yet untitled, is being written for the beginner.

All games are text-only and will be available for the Spectrum 48K.

Sorcerer on the way

THE SECOND game in Infocom's fantasy series, Sorcerer, is in the pipeline.

The follow-up to Enchanter, which went on sale late last year, Sorcerer is scheduled to follow the US release on March 15.

The game begins where Enchanter left off: Belboz, the necromancer, your friend and mentor, has vanished and foul play is indicated. The freedom of the land is indicated. The freedom of the land and the survival of the Circle of Enchanters are at risk. Your task is to rescue the necromancer and save the kingdom.

In contrast to the Zork trilogy, the fantasy series centres on magic rather than exploration and treasure hunting, according to Infocom.

The product manager for Infocom, Michael Dornbrook, said Sorcerer players will advance toward their goal through the use of magical powers that they acquire during the adventure.

Michael said that Sorcerer understands a vocabulary of more than 1,000 words, "nearly 50% more than any other interactive fiction product.

"That capability allows Sorcerer to add substance to the Zork-Enchanter universe by supplying it with a richly detailed history and geography," he said.

Sorcerer was written by Steven Meretzky, the author of Planetfall. Infocom games are available for the Atari, Apple, Commodore 64, IBM PC and the TI 99/4a.

Forest launched for CBM 64



AUDIOGENIC claim to have taken the Commodore 64 into another dimension with their new game Forbidden Forest.

It is one of the new generation of disk-based games and features not three, but four dimensions to accommodate time.

The adventure takes place in a scrolling forest "which many have entered, but none have returned", says the company.

The hero is a champion archer whose task it is to find the evil Demogorgon who is hiding somewhere in the vast forest.

The archer has to battle giant spiders, skeleton soldiers, snakes, frogs and dragons before he meets Demogoron in a duel to the death. After which he becomes the victor.

The graphic adventure is available from the usual outlets for £12.95.

Dragon software rereleased

TWO Dragon Data games, Calixto Island and Black Sanctum, already available in text versions, will be rereleased as animated graphic adventures soon.

Also on the way are two new games called Sea Quest and Shenanigans.

Sea Quest involves searching for buried treasure and in the quest in Shenanigans you have to follow the rainbow to find the pot.

Finally, Carnell Software's Wrath of Magra, seems to be living up to its name, even before its launching.

Production problems meant that the release of the game was postponed from mid March to mid April.

VALH

GAME OF THE YEAR
BMA'84

Available Now for Commodore 64

ORCS BEWARE!

YOU'VE LOVED THE BOOKS

YOU'VE PLAYED THE GAMES

NOW READ THE FABULOUS MAGAZINE



WARLOCK

THE FIGHTING FANTASY MAGAZINE

* Cash prizes for your own fantasy adventures
* 'Out of the Pit' background features on monsters, mapping, *and* Allansia *the* fighting fantasy kingdom
* A **new** WARLOCK OF FIRETOP MOUNTAIN Quest



FIGHTING FANTASY

ISSUE ONE **OUT NOW** 48 pages 95p

GETting things moving

Andrew Pepper *describes the program techniques that allow a player to move objects in an adventure*

VIRTUALLY ALL adventure programs allow the user to manipulate objects. The programs have to provide facilities to GET or DROP objects. But special facilities and techniques are associated with moving objects.

In adventure programs the user can GET an object, move to some other room and DROP it there. The user can learn what objects are being carried by requesting an inventory. There are some variations about this basic scheme. I will describe them but I will not explain how they are implemented. They are a complication which would only obscure the points I am trying to illustrate.

There may be some objects which cannot be moved. The program has to perform checks when the user tries to get such an object and tell the user that the object is impossible to move. When the user requests to get an object the system should check that the user has not entered the name of a monster or some other person in the adventure. If this is done then the following type of situation is avoided:

You are in a magical hall.
There is a wizard here.
Get wizard.
I can't see that.

The system has apparently suddenly forgotten about the wizard. If the system checks that the object requested actually is an object then the dialogue would be more like:

You are in a magical hall.
There is a wizard here.
Get wizard.
You can't get people!

One problem that will be described in more detail concerns the type of object. Suppose you are writing an adventure and have placed some gold and a knife in a room. If you are not careful the room might be described like this:

There is a knife here.
There is a gold here.

You could avoid the problems by describing the gold as "a gold bar", however I recommend implementing a more general scheme: for each object in the adventure the name of the object is stored. The first character of the name is a number which indicates the type of description to use for the object. For example the knife is stored with the name Oknife. The system will describe this as a knife. Gold is stored with the name Igold and this will be described as some gold.

Having decided the type of facilities we want to have within the program we can start to design the computer code. When describing the code I will only use the so called ANSI BASIC.

I will store the names in a string array, N$(). The first character of the name will indicate the type of description to be used for the object. As well as this string array I will also have a numeric array N() to contain the room number in which the object is held. To indicate that the object is being carried by the user then we will set the room number to −1.

Four commands

There is an initialisation section, lines 10 to 130. This sets-up the objects in the adventure and initializes the phrases, in e$ and d$, which describe particular objects. The current room number is given and any objects in the room listed. Instead of writing a full parser to decode an English phrase entered by the user, the system merely displays the four commands which the system accepts and requests the one required (lines 140 to 200).

The MOVE command (lines 220 to 250) requests and obtains a new room number. In a full adventure program the user moves around using compass points. The GET command (lines 270 to 330) checks that an object exists, checks that the object is in the room and, providing all is well, GETs the object (changes its room number to −1).

The DROP command (lines 350 to 410) checks that an object exists, checks that the object is being carried and, providing all is well, DROPs the object (changes its room number to the current room).

The inventory command (lines 430 to 520) lists all the objects which has as a room number −1. If there are no objects with this room number then the user is carrying nothing. Note in particular the code in lines 480 and 490, which decide ▷

```
10 REM ##### demo adventure - moving objects
20 REM
30 DIM N$(10),N(10),D$(10)
40 RESTORE 690
50 B = 6
60 FOR I = 0 TO B
70 READ N$(I),N(I)
80 NEXT I
90 D$(0) = "is a": D$(1) = "is some": D$(2) = "are some"
100 E$(0) = "A": E$(1) = "Some" : E$(2) = "Some"
110 REM
120 REM #### initialization complete
130 R = 1
140 GOSUB 590 : REM ### describe room
150 PRINT "Move, Get, Drop or Inventory";
160 INPUT A$
170 IF A$ = "" THEN 150
180 A$ = LEFT$(A$,1)
190 A = ASC(A$)
200 A$ = CHR$(A AND 95)
210 IF A$ <> "M" THEN 260
220 PRINT "Which room number";
230 INPUT R
240 IF R < 1 OR R > 10 THEN 220
250 GOTO 140
260 IF A$ <> "G" THEN 340
270 PRINT "Enter object name";
280 INPUT A$
290 GOSUB 530 :REM  see if object exists
300 IF I = 99 THEN PRINT "I can't see one of those!" : GOTO 140
310 IF N(I) <> R  THEN PRINT "I can't see that here!": GOTO 140
320 N(I) = -1
330 PRINT "OK": GOTO 140
340 IF A$ <> "D" THEN 420
350 PRINT "Enter object name";
360 LINPUT A$
370 GOSUB 530 : REM see if object exists
```

An example of an adventure that incorporates moving objects

```
380 IF I = 99 THEN PRINT "I don't know that object": GOTO 140
390 IF N(I) <> -1 THEN PRINT "You are not carrying that": GOTO 140
400 N(I) = R
410 GOTO 330
420 IF A$ <> "I" THEN 140
430 PRINT "You are carrying:"
440 C = 0  : REM c used to count objects
450 FOR I = 0 TO 8
460 IF N(I) <> -1 THEN 500
470 C = C + 1
480 A = VAL(LEFT$(N$(I),1))
490 PRINT E$(A);" ";MID$(N$(I),2,99)
500 NEXT I
510 IF C = 0 THEN PRINT "Nothing"
520 GOTO 140
530 I = 0
540 IF LEFT$(A$,4) = MID$(N$(I),2,4) THEN RETURN
550 I = I + 1
560 IF I <= 8 THEN 540
570 I = 99
580 RETURN
590 PRINT
600 PRINT "You are in room";R
610 PRINT
620 FOR I = 0 TO 8
630 IF N(I) <> R THEN 660
640 A = VAL(LEFT$(N$(I),1))
650 PRINT "There ";D$(A);" ";MID$(N$(I),2,99);" here"
660 NEXT I
670 PRINT
680 RETURN
690 DATA "0knife",2,"0gun",1,"1gold",6,"2silver bars",8,"1food",9,"0bottle",7
700 DATA "0lamp",5
```

◁ how the object is to be described in the adventure.

Lines 530 to 580 are a subroutine which checks if the object A$ exists. If it is amongst the list of objects then the object number is returned in i. If the object does not exist then i is set to 99. Lines 590 to 680 display the room number.

You will almost certainly be able to produce a shorter version of the program if you write it specifically for your machine. When run the program will show: You are in room 1.
There is a gun here.
Move, Get, Drop or Inventory?

Type the initial letter of the command which you want. For example if you want to get the gun then type "G". Your screen will show: Enter object name?
Enter "gun". The system will show:
Enter object name? gun
OK
You are in room 1.
Move, Get, Drop or Inventory?
Note that the gun has disappeared from the room. If you request an inventory, by entering I, then the system will tell you that you are carrying the gun.

There are other things to note. The descriptions are all consistent. The system will say "There is some gold" or "There are some silver bars" as appropriate. It is possible to get objects from one room, move to another room and drop it.

As I indicated there are some extensions which could add to the example programs. For example it would be quite easy to place a limit on the number of objects which the user can carry. However the example program is quite close to the real programs that are written. □

Unravelling Valhalla

***Mike Stackwell** puts Valhalla under the magnifying glass, being careful not to reveal critical clues*

WITHOUT DOUBT, Valhalla is very complex, both as a game and as a program. Preceded by many mysterious adverts, it was finally released on October 2 last year to much critical acclaim. Since then it has brought in around £2 million worth of sales to its publishers, Legend.

Based loosely on a Norse legend, it depicts Asgard, the mythical home of the gods, which is surrounded by two other areas, Midgard and Hell. The aim of the game is to collect six quest objects and thereby enter Valhalla, which was the Norse equivalent of paradise.

The problem when writing articles about adventures is that it is all too easy to give information away that spoils the adventure for those people who haven't got as far as the bit you're talking about. So, although I will not reveal any major secrets I will mention facts that some people may not wish to know (yet). You have been warned.

The world of Valhalla consists of 81 locations, all unique, although some may appear very similar to the unwary explorer. The map is nine by nine in size, and has the unusual characteristic that leaving a location in one direction means that you will enter the next from the opposite way. For example if you exit to the west you will enter the next place from the east. This of course holds true for all of the directions, as does the fact that if you go back the way you came you end up where you came from (if you see what I mean).

Although this may seem obvious, Valhalla is the only adventure I have played that holds rigorously to these rules. However, in common with other adventures, some routes are one-way only, although again this is not as frequent as one might expect.

Starting line	Action
6000	Pause
6050	Save
6100	List
6150	NOT USED
6200	Who*
6250	Load
6300	Print
6350	Noprint
6400	Copy
6450	What
6500	Where
6550	Help

*It is not advisable to use "who" for a break-in routine, as, providing a command is not waiting to be executed, the program does a "who" automatically when you enter a location

A list of utility routines

Some areas can only be reached if you are carrying a certain quest object, or if you have somebody with you. The opposite can also apply — you may have to drop an object or tell someone to go away, though not too rudely unless you want a visit from Mary.

Each location can hold no more than six objects, including chests/cupboards and any items in them. To prevent this rule being broken, if at any time an extra object is dropped in a location that already has six in it, the aptly-named Klepto comes in and steals the surplus object.

Finding Ofnir

Unfortunately there is a bug in this routine — try putting six objects in a chest. This equals seven objects altogether, so in comes Klepto . . . and the program crashes, usually with a Subscript Wrong error, although odder effects have been known. The same, of course, applies to cupboards.

Just under half (38) of the total number of locations in Valhalla can be reached without having found Ofnir, and for the sake of secrecy it is mainly this part of the world that I will be investigating. The accompanying map shows the whole of Valhalla, but with only those 38 locations mapped out. Of course, Ofnir is in there somewhere, but I'm afraid that you'll have to find it yourself. Here's an additional clue to the one in the manual: head north boldly, but remember that giants may get in the way. Once Ofnir has been found, this allows access to several new locations, and so to Drapnir . . .

One common problem seems to be how to get out of El Vinos, whose distinguishing features are a very tightly locked chest full of wine and a resident drunk, the infamous Mary, who is in possession of no less than six bottles of wine and yet steadfastly guards a chest that she can't open.

There is no obvious way out and no ringway, and although the HELP routine assures you that you can go N, NE, E, SE or NW (an apparent bug which I shall mention presently) it took me a very long time to suss out that all you have to do is FFO TEMLEH RUOY EKAT (read backwards) before attempting to exit. Similarly you cannot go north from location 51 (see map) if you are carrying an axe, though since I always choose a sword this has never particularly bothered me.

Valhalla (the program that is) comprises 20K of BASIC, 7½K of variables which are added to as the program is run, and about 12K of machine code. Initially I was quite surprised to find so much BASIC involved, but I think that Legend were quite justified in using it.

The machine code handles the actual drawing of the graphics, and the visual movement of the characters, as well as scanning your input for recognisable keywords (eat, drink, throw, etc). It also handles the keyboard via its own vectored (IM 2) interrupt scheme, thus allowing you to type in up to two commands while action is proceeding on the screen.

All that the BASIC is left to do is actually run the game, acting on information from the player (via the machine code) and outputting messages and graphics (both again done in machine code). This, of course, involves a lot of logic to ensure that characters don't overlap each other on the screen (it doesn't always succeed) or that when Mary comes in to thump you for misbehaviour she isn't already on the screen. (Alcoholically induced double-vision perhaps?) It also controls the results of fights, eating and drinking, in fact most of the behaviour that you see.

The BASIC part of the program is easy enough to examine, for the program is not protected — rather surprising, considering the price of it — though this doesn't seem to have done Legend's profits any harm.

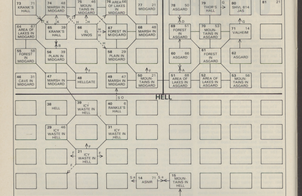
In my opinion it is well written, although there are bound to be people who will disagree. The programmers have made extensive use of user-defined functions and logical conditions for simplification, and is quite easy to follow once you realise what's going on and bother to look up the definitions of some of the more commonly used functions.

A puzzle

To save memory, variables are used instead of some of the more commonly used numbers (thus saving six bytes every time one is used) eg o=0, l=1, z=2, k3=3, k4=4 and so on up to 10, where it goes 10, 11, 12 up to z0 (20). There are also others for such numbers as 255, 39 (the number of people in the adventure) and so on.

All the words used in the adventure are stored in an array, d$(255,11), although for technical reasons too long to go into here, not all of the array elements are used.

The first character in each entry is the CHR$ of the length of the word stored in the next nine characters. The last character in each entry is a bit of a puzzle, but I am reasonably sure that it indicates ▷



The Valhalla map which shows the three main regions included in the game with explanations below

1. The number in bold in the top-left of each box is the location number, the number in the top-right is where the ringway (if present) leads to
2. A solid line indicates that you can travel freely along it (except where indicated by a one-way arrow)
3. A dotted line means that you can only travel in the direction shown provided you have satisfied the condition indicated by the accompanying letter (see below)
4. Some routes are conditional in one direction but not in the other (eg 69-78)
5. Only locations accessible without any quest objects are shown in full.

Key: A Must not *be carrying axe, B Need Boldir present, D Need Odin present, F Must be carrying Ofnir, G Must be carrying Grimnir (!), H Must* not *be carrying helmet, K Must* not *have Krank present, Sa Must be carrying Skalir, So Must be carrying Skornir.*

◁ to the system whether the word is a noun, verb or proper name. There are, not surprisingly, several user-defined functions for extracting words from this array. The main ones being FN w$(n), which returns a string containing the nth word in the array, cut down to size, and FN b$(n) which works like FN w$ but adds "a" or "some" in front of the word as appropriate, eg "some food". Altogether there are 165 words that will be recognised, including 14 swear words.

As far as I can make out everything else in the program relates back to this array. There are two other arrays of interest, c$(39,20) and m$(81,18) which holds information about 39 characters.

Although I have successfully investigated both, and even written programs which print complete descriptions of every character and location in the game world, I am rather reluctant to divulge the methods as it is all too easy to find out where the quest objects are kept (though not how to get to them) and it spoils the fun for those who, like me, love dredging through other people's programs. To start you off though . . . as previously mentioned everything is vectored through d$, and so the 39 characters (not steps) are arranged in the same order as their names in d$, which just happen to run from d$(1) — "you", to d$(39) — "Raven".

Similarly the objects that you (or anybody else) are carrying are represented in c$ by their position in d$, eg "ring" is d$(85) and so if Odin is carrying a ring, it appears somewhere in c$(3) as CHR$ 85, "Odin" being the third element of d$. C$ also holds your attributes, ie good/badness, charisma, strength, bravery, brains, and a sixth which I call, rather unoriginally, "the 'X' factor". If anybody knows what it really is then please write in and let me know. Also in C$ is your current (as opposed to maximum possible) strength.

Altering the latter means that you can choose your start position — it could even be Valhalla itself, although that would take the fun out of it. Strangely enough, although your exact position on the screen is also saved in c$, the direction in which you're facing is stored in an array d(39) – for the 39 characters. As far as I ▷

EMPIRES

When a Galactic war breaks out, there is only one winner – make sure it's you!

Empires is a game of the type which has not been seen before. It is a fully player interactive strategy game for up to six players and an umpire. The basic set contains three players. Expansion set one contains three more. Each player controls a unique race struggling to take over the galaxy. Mining companies, space fleets and sound strategy all contribute to the success of your race; but beware the other players will be trying to tear your empire apart. The skill required to organise your empire as it grows is enormous. To assist you galactic maps and data cards are provided. Soon further expansion sets will be available to give you an advantage over the other players. Move information is transferred via cassette to the umpire who analyses it with his program and returns the new data to the players. Messages can be sent to other players in the same way.

THIS IS A NEW CONCEPT IN STRATEGY GAME PLAYING.

48K SPECTRUM

CONTENTS: THREE PLAYER CASSETTES, ONE UMPIRE CASSETTE, 4 GALACTIC MAPS, 81 DATA CARDS, 1 FULL SET OF RULES, 3 PLAYER INSTRUCTIONS

Price £19.95.

Please send me .... copy/copies of Empires Basic Set

Total cheque/P.O. enclosed ....
Cheques payable to Imperial Software.

Send order to:
IMPERIAL SOFTWARE
IMPERIAL HOUSE
153 CHURCHILL ROAD
PARKSTONE
POOLE
DORSET

Name ....

Address ....

.... Age ....

◁ see this array serves no other purpose, being full of ones (=right) and zeroes (=left), and therefore apparently wasting about 156 bytes.

Having gone rather overboard on c$, I shall say little about m$, except to say that it works on the same principles as above. And of course, the fact that you can only have up to six objects in a location should help happy hacking.

Half truths

Bugs: apart from the Klepto one mentioned above, I know of only two others. On the first 15,000 or so copies made, Felstrong, the fifth quest object, can in fact be discovered before Ofnir. The other one is rather more annoying. When a route is conditional (eg You need Odin's help if you wish to go north). The Help routine appears to misbehave. The message stays the same whether Odin is there or nore. Yet other locations (El Vinos) state that you can go north (or wherever) but in fact you can't until you have dropped a certain object, or got somebody to leave.

Legend may claim that (page 25 of the manual) "Help does not always tell the 'whole truth', but will not actually lie". I suspect that there is a bug since inspection of the HELP routine (lines 6550-6595) reveals lines such as "FN w$(o2)+' must not be present'". And yet whether the person or object in question is present or not it just says "you can go north."

Once the program is running it is difficult to stop it and study arrays or make alterations because of the use of IM 2 and the fact that POKE 23659,0 is used so it is best just to set initial conditions. However, if you are willing to sacrifice the use of HELP or a similar function (see the utility list) then insert a line such as:

```
6550 POKE 23659,2: POKE 23296,237:
POKE 23297,86: POKE 23298, 201:
RANDOMIZE USR 23296: STOP
```

This will return you to BASIC when you type HELP. But beware, Valhalla seems to do strange things to the BASIC error-handling system: if you (like me) type Shift-Break in reply to "scroll)" then the system will crash. To restart, whether from my routine or any other stoppage, simply type GOTO 9800. On a similar line it is also possible HELP give answers all the time (except when there's nothing to say) by modifying line 6550, which contains a rather obvious RND . . .

One final subject: microdrives. Contrary to what you may have read elsewhere, it is quite easy to modify Valhalla for drives, which apart from the obvious speed increase, also makes experimentation easy. The only slight problem is that for some reason best known to themselves, Legend put the LOAD, SAVE, and scroll text window up routines in a REM statement, rather than in the main block of machine code.

In theory you should alter the four relevant USR statements to take into account that PROG might move. In fact, providing you do not start opening files or RS232/net channels, you should be quite safe with the following alterations:

```
5010 . . . RANDOMIZE USR VAL "23818" . . .
5016 . . . RANDOMIZE USR VAL "23818" . . .
6050 . . . RANDOMIZE USR VAL "23850" . . .
6255 . . . RANDOMIZE USR VAL "23877"
```

Once these changes have been made, all you have to have to do is SAVE the various parts of the program (not forgetting to alter the BASIC loading routine) onto microdrive, SAVEing the machine code as CODE 53350,12185. You must only LOAD one part at a time into memory when doing this, or you will get an "Out of memory" error when you try to MERGE in the BASIC.

Unfortunately I haven't yet converted LOAD and SAVE themselves for drives since it is really quite a complex job. In fact it is probably not feasible because of the extra memory occupied by the program when it is running.

Fluttering flags

I hope the hints I have given you will be useful and that you don't use them to indulge in too much cheating. Besides, at the time of writing (March 24) I hear it has just been solved by somebody — he must be in ecstasy.

PS Have you noticed how the flags on the castles flutter in the breeze as you type? Watch carefully next time you play. □

Finding the right role for your micro

Gren Hatton *looks at how a computer contributes to realism in role playing games*

IN APRIL the concept of using the computer to control or supervise certain aspects of role-playing games (RPGs) was introduced. The next step is to create a number of separate program modules to look after each of these aspects, and to develop means of interfacing between one program and another by use, say, of disk back-up.

Some of the principles which I have found useful when setting out to design a specific program follow. It is always best to demonstrate techniques by referring to examples from a particular program. The vehicle I have chosen is a program called Return to Thongor, which I put together in an attempt to get around some of the difficulties experienced by my own RPG group during two particular stages of every game.

Last month I described the six stages of an RPG. These were: creating the characters; providing the background data to start an adventure; journeying to the point where the adventure is to take place; the main action of the adventure; the journey back to base camp; sharing out the proceeds of the adventure, and general stocktaking.

Experiment

I had devised this program as an experiment to help add realism and speed up play at stages two and six of the adventure. Here the common element is likely to be that both stages take place at the base camp or home territory of the party of adventurers. The program provides a series of one-player adventures in the City of Thongor, a made-up medieval scenario which we use as our operations base when adventuring.

The program is, in effect, an attempt to create a city, complete with shops in which you can buy and sell a variety of goods. It has taverns, in which you rest, drink and meet other adventurers and characters, a castle with guards, a password and dungeons, many passersby, some of whom are aggressive, some helpful, others too busy to stop and a variety of specialists, such as guildmasters, sages, mapmakers, river- and sea-pilots, seers etc, who can be consulted for special help.

The aims are threefold: a) To give players a means of buying new gear or selling off surplus gear from an adventure. b) To provide a rich and varied source of gossip and rumours as a basis for new adventures. c) To allow solo adventurers to carry on their own separate branch of the main game on auto-pilot, while the DM carries on dealing with the rest of the party.

The easiest way for me to write about the program is to describe the concepts involved and to offer a few words of advice on how to write programs of this type.

Features

The first job is to jot down all the features you can think of (in any old order) which you would like to have in the program. For instance: buying and selling, meeting in taverns, encounters with the city police (we call them proctors), having your pocket picked, descriptions of the inside of shops, being set upon by robbers, trying to get past the castle guards, having old or worn weapons repaired, a magic-vendor's shop which is invisible to all those who have no magic powers, a game-time clock which regularly prints out the time of day or night in the game, finding that a shop is shut, discovering that you have no money left, or cannot afford to buy some item which you want, overhearing some gossip (which may be either vital or completely worthless!)

When you have filled a few pages with scribble and crossings-out, look long and carefully at the list. The items on it will gradually start to sort themselves out in your mind into two main categories listed below.

List A: main events of a generalised nature, such as are likely to be repeated in any adventure. For example going to a shop, popping into an inn, visiting the castle, stopping a passerby, going to visit a specialist.

List B: subsidiary events which may apply equally to several of the main events by way of added description, eg description of the interior of a room, statement of the time of day, statement of how many gold pieces you are carrying, suddenly finding that your pocket has been picked, running into a patrol of proctors and so on. List A will probably be much shorter than List B. If it isn't, try again. As a broad guide, you can make the items on List B into subroutines, and the items on List A will become the main action sections of your program.

Having got this far, draw up a chart which links all the items on List A in a continuous chain. There are many ways of setting about this, such as making a flow chart of decisions and possible responses. But a good approach to this particular type of essentially text-based program is the menu system, so called because of its resemblance to a restaurant menu.

For example, "Will you 1) Stop a passerby? 2) Visit the castle? 3) Go shopping? 4) Go to a tavern? 5) Visit a specialist? 6) Return home? Enter (1), (2), (3), (4), (5) or (6)".

Obviously, at the end of each of the five main sub-sections there will be a GOTO instruction returning you to the menu to make another selection. Whilst choosing (6) takes you into an exit routine which reports your final status (gold pieces on hand, time of day or night etc) and ends the program.

Next you should set about structuring your program. Don't be alarmed, this is just a programmer's buzz word. There is nothing magic or hard to understand about it. All it means is that we set out the program, using in this case our menu as a guide, and assign blocks of line numbers for the various sections.

After all, you will not just sit down and write the program from A to Z in apple-pie order. It will be done one chunk at a time, and the first bit which you get working on its own might well be a section right in the middle, or one of the parts which will ultimately be right at the end of the listing.

Think of those parts of the program which you already begin to realise will be lengthy, and allow a bigger block of line numbers for them. Taking the example given above, we might produce something like the accompanying diagram.

Spaces

Leave one or two big spaces in the structure if you can. You will almost certainly want to make some additions or do some shifting later on. Now you are ready to tackle a section of the program. Be prepared for heartache and eyestrain, and try to get to bed no later than 2am.

I can make some suggestions here, based on my own experiments.

- Start off with the bare bones of each section of the program and get it running without any wordy descriptions.
- Next add some gilding. Our friend the Random Phrase Generator, which I described in principle in the last issue, can be brought in time and again as a series of subroutines to deal with such things as random descriptions of the interiors of shops and taverns; travel through the city from one location to another; the number of people in the streets; rooms, ▷


BEYOND
CHALLENGING SOFTWARE
Less than human, far more than mere computer
The Psytron controls the massive Betula 5 Installation. When the attack comes, it will cope with defensive demands which would leave a human brain unhinged, computer circuits scrambled.
Damage in any sector of the base must be assessed and its effect on the fabric of the installation calculated immediately.
Human lives will be expended as necessary but if the Psytron ever goes down.
The Sinclair Spectrum has always attracted the best programmers. It's always had the most innovative software: Psion Scrabble, The Hobbit, Jet Pac and now . . .
PSYTRON
Ten screens of stunning graphics, animation which makes other programmers gasp, a 20 page booklet of hints, tips and help with the discovery of each of six levels of play.
"How long can you survive the Final Conflict? A true Psytron would last an hour and a Sinclair Q.L. goes to the first person to match that feat."

INITIALISE VARIABLES AND WELCOME TO PROGRAM	LINES 1-100
STOP A PASSERBY	LINES 200-500
VISIT THE CASTLE	LINES 1000-1999
GO SHOPPING	LINES 2000-3999
GO TO A TAVERN	LINES 4000-4999
VISIT A SPECIALIST	LINES 5000-5999
EXIT ROUTINE	LINES 7000-7499
SAVE GAME ROUTINE	LINES 7500-7549
GOSUBS BLOCK	LINES 8000-9999

Program structuring by assigning blocks of line numbers

◁ passages in the castle, and so on. This will give variety and apparent spontaneity to any events which you have to call up regularly in the program.

● Add, in each section, a number of possible branch points, where you have a random chance of encountering one or more of a series of events stored as subroutines. You could for instance:

Meet a party of proctors
Be accosted by an old beggar
Have your pocket picked
Overhear a rumour

Store

Let us say that these are stored as subroutines starting on Lines 8500, 8600, 8800 respectively. You could add a section in your program just at the point when you are about to enter a shop or a tavern (say at Line 2200), something like this:

```
2200 IF INT(RND*8)=2 THEN GOSUB 8500
2202 IF INT(RND*5)=2 THEN GOSUB 8600
2204 IF INT(RND*20)=2 THEN GOSUB 8700
2206 IF INT(RND*8)=2 THEN GOSUB 8800
```

Thus, each time the program brings you to a shop door or a tavern door, you have a 1-in-8 chance of meeting a patrol of proctors coming round the corner, a 1-in-5 chance of being pestered by a beggar, a 1-in-20 chance of having your pocket picked, and a 1-in-8 chance of hearing some rumour or gossip. A little ingenuity and experiment will teach you how to use this technique to create an extremely spontaneous scenario with a very high degree of realism.

● Use the type and level of your characters as triggers which will evoke different responses from the permanent characters such as innkeepers which you may have built into the program. For instance, a thief might have a 1-in-20 random chance of being recognised by a shopkeeper and thrown out of the shop, or a 1-in-30 chance of being chased by the proctors. Again, a shopkeeper might run to open the door for a hawk-eyed and commanding fifth-level fighter, but keep a lowly first-level character waiting in line to be served while game time ticks remorselessly away.

● Introduce a game-time clock. You need a simple subroutine to make the clock run:

```
8400 REM clock mechanism
8402 IF minute>59 THEN LET minute = minute — 59: LET hour = hour + 1
8404 IF hour>23 THEN LET hour = hour — 24: LET day = day + 1
8406 RETURN
```

Now you can introduce time jumps at specific points in the menu. For instance, each time you decide to enter a shop you can add a line —

```
LET minute = minute + 30: GOSUB 8400
```

This will have the effect of making you spend 30 minutes in the shop and automatically updating the clock.

If you want to be more realistic, you can have several mini-jumps in this part of the program — say a five-minute jump for entering the shop, another 10 minutes if you have to stand in a queue, and a further five minutes each time you buy or sell something. Of course, you also need to sprinkle several time-display instructions at strategic points in your routines:

```
2010 PRINT "The time is now "; hour ; ":" ; minute
```

followed by

```
2012 INPUT ("Is it time that you returned ? "); z$
2014 IF z$ = "n" THEN GOTO (a line which continues this part of the program)
2016 IF z$ = "y" THEN GOTO (a line which gets you back to the main "menu")
2018 GOTO 2012 (a safety precaution to reject any answers other than "y" or "n")
```

You can use this technique throughout the program, to cover time spent in shops, taverns, waiting about on street corners, moving from one part of the city to another and so on.

● Once you have a game-time clock in the program, life can be made much more interesting. You can also start to think of interactive subroutines to a much greater extent. This mouthful, like the random phrase generator and the structured program, sounds daunting but it's only jargon.

Consider the variable hour which we have used in the game-time clock subroutine above. We can have another subroutine called REM time to go home, which deals with the fact that we have to be back home by midnight, let us say. Suppose this is stored in the subroutine block of our structured program at Line 9000. Then when we update the clock we can say,

```
IF hour = 24 THEN GOSUB 9000: GOTO (a line which gets you home)
```

Try the effect of inserting this in the example above as Line 2011

To take a different kind of example to illustrate the same point, consider the random phrase generator. We may be using it to describe our passage from one part of the city to another, as mentioned above, and in so doing we may include a phrase such as:

```
9500 PRINT "The streets are crowded"
```

Clearly, even in a busy place like Thongor City the streets are hardly likely to be full of life 24 hours a day. But with hour as a constantly updated variable in our program we can replace this bald statement by a separate subroutine where we can store a set of alternatives.

```
9500 IF hour>5 AND hour <8 THEN PRINT "There are already one or two folk about"
9501 IF hour >8 AND hour <10 THEN PRINT "The streets are now quite busy"
9502 IF hour >10 AND hour <17 THEN PRINT "The streets are crowded"
9503 IF hour >17 AND hour <19 THEN PRINT "The streets are less busy now"
9504 IF hour >19 AND hour <21 THEN PRINT "The shops are mostly closed, and the streets quite empty"
9505 IF hour >21 THEN PRINT "The darkened streets are filled with shadows"
9506 IF hour <5 THEN PRINT "At this hour, only thieves and footpads disturb the night"
9507 RETURN
```

Stick in plenty of alternatives. You can afford more program space for this since, as a subroutine, you can use it again and again by calling on it from other parts of the program or from other subroutines.

Interaction

Another example of a subroutine designed for interaction with others is the gold piece monitor. It is a simple subroutine which prints out a reaction depending on the amount of money in your purse. For example:

```
8900 REM gold piece monitor
8901 IF gold = 0 THEN PRINT "I have no money at all!" : RETURN
8902 IF gold = 1 THEN PRINT "I am reduced to my last gold piece" : RETURN
8903 IF gold >1 AND gold <10 THEN PRINT "Oh dear! I only have"; gold; "gold pieces left" : RETURN
8904 PRINT "I now have "; gold; " gold pieces" : RETURN
```

Any time that you have to buy or sell, or if you get robbed or have to pay a toll or a fine, add in a couple of lines which bring in the above subroutine like this: for buying, having identified the cost of some item as the variable cost,

```
2500 IF cost >gold THEN PRINT "That's more than I can afford!" : GOTO (some line which returns you to a point where you can choose again)
2502 LET gold = gold — cost
2504 GOSUB 8900
```

For selling, having identified the value of your saleable goods as the variable val,

```
2800 LET gold = gold + val
2802 GOSUB 8900
```

If I am not to make a long book out of this introductory article, I must eventually cry enough and stop quoting examples. ▷

◁ The list is endless, and many people reading this will already have thought of a number of ideas to try out for themselves. It might help to make a short summary of the main ideas.

- Seeing a number of programs as complementary and therefore using the same variables for the same purposes in each, to enable program modules to be linked by a disk-access system.
- The use of a menu within a program module, to cater for "what-shall-I-do-now?" situations.
- Structuring of the program listing within each module so that you do not end up with a program which is virtually impossible to understand or debug.
- The concept of the random phrase generator, to add apparent realism and spontaneity to dialogue and descriptions.

Update

- Use of the string array to store data which lives in the program as a quantity which is constantly referred to or updated, or which can act as a conditional trigger to initiate a variable response, for instance, the contents of a map or player or party characteristics
- Intelligent use of a wide variety of subroutines.
- Interaction between subroutines, for example, weather, clock and time-dependent descriptions, hitpoint monitor, gold-piece monitor in taverns and shops.

Questions concerning the future on this topic are dynamic. It is so easy to say "Well, there won't be anything new in the way of ideas on this subject . . ." or "Of course, we're stuck with the type of hardware on the market at the moment..."

Realism in computer-based games must draw increasingly heavily on such concepts as the Random Phrase Generator to allow the computer to make its responses with more apparent spontaneity. Clearly, such techniques, even if compressed to the minimum, still require significant amounts of RAM.

At the same time, there is a limit to the amount of RAM available in home micro's. Currently about 48K in today's 8-bit machines, and perhaps between 128K and 256K in the next generation of 16-bit models, which no doubt will very soon appear.

We are in a sort of no-man's-land at the moment. The shape of things to come may have been revealed in Sir Clive Sinclair's QL. But its back-up storage, the halfway-house, tape-based microdrive, is a bit on the small side at 85K a throw to give sensible back-up support of the type needed for text-based adventure gaming of the type I have described.

Perhaps the next step will be the advent of a much wider choice of 1Mb disks in the £150-£200 range, so that we can see our way forward confidently in starting to hang really large storage on to a games machine in the home.

At all events, it seems to me that the way forward is likely to involve building up a series of independent but related games-aid modules, similar to those I have described. To get each module working in a modest RAM store, 48K is the obvious choice at present; then, by intelligent use of common-denominator variables, to set about making the various modules interact with each other using disk-dumping and retrieval procedures.

Collaboration

It would be interesting to know if other people have similar views and similar aims. This is a subject which immediately suggests collaboration to achieve its ultimate form. However, some basic rules of conduct should be set down to avoid disaster:

a) Define the overall structure of the set of programs and make sure that the interface between modules is very carefully agreed.
b) Agree on the names of variables that will be used in each program, particularly where the contents of a variable store will be carried from one program module to another.
c) Agree on the protocols for carrying variables from one module to another and retrieving them again.
d) Avoid the use of programming techniques or commands which are too specific to a particular machine.

(Any readers interested in further information about programs mentioned in this article should write to the author at Stoneleigh, Middle St, Kilsby, Rugby, Warwicks — enclosing a large SAE. □

SOFTWARE INVENTORY

What's on the way in the adventure world — if you have a new adventure, war game or real-life simulation which you are about to release send a copy and accompanying details to Software Inventory, Micro Adventurer, 12-13 Little Newport St, London WC2R 3LD

A republic gone bananas

Strategy *Banana Drama* **Micro** *Commodore 64* **Price** *£9.95* **Format** *Cassette* **Supplier** *Visions Software, 1 Felgate Mews, Studland St, London.*

WITH A title like Banana Drama, I though this game would be concerned with taking three girls, cropping their hair, giving them some awful songs to sing and then making a quick million bucks.

The truth is not quite like that (is it ever?) and the action moves over to a banana republic. As president of Ananaba (almost a backwards banana), it is your job to look after this "small but important independent state" for as long as you can remain in charge.

On loading this purely tape-based program has an annoying habit of trying to access the disk drive, if you happen to have one connected and switched on. Why? Who knows.

Like most strategy games, your success or failure depends on making the right decisions at the right time, and like most good strategy games there are a lot of decisions to make.

This involves single key presses, and while you make up your mind whether to give the USA oil exploration rights of not, the computer will annoyingly bleep-bleep at you all the time. This happens at every decision-making point, and the sound soon got turned off.

To keep yourself in power for as long as possible, you have to remain on good terms with a large number of people on whom you depend for information.

For instance, on Ananaba the State Security Organisation will, for a fee, give you a quick report on the possible consequences of any action you take. At least, they will as long as you manage to stay in their good books. I soon fell out with them and was reduced to moving around in the dark, alienating myself to everyone under the sun, until I decided to make the State Security Organisation omnipotent. In retrospect, this was not a wise move.



You have subversives to deal with, as well as trade unionists, plantation workers and owners, and other run-of-the-mill Ananaba people, and all these must be kept happy for as long as possible.

You must also balance the books of the exchequer as well. Here it helps to be on good terms with foreign powers, since the Kremlin and the White House are always good for a few roubles or dollars if you're in their good books. The USSR proved itself to be more generous than their American counterparts on one occassion, but then I was selling them cheap bananas at the time.

Eventually everything came to a halt after 15 months at the top when I was assassinated by trade unionists, having already beaten off an attempted coup by the subversives. Just because some people don't get recognised political status, they think they can overthrow anybody.

An interesting and good game, and if you like this sort of thing, Banana Drama should keep you amused for quite some time. Some of the music is a little, well, not quite right, but everyone's entitled to slip on a banana skin sometime. **PG**

A challenge to foil enemy

***Simulation** Stonkers **Micro** Spectrum 48K **Price** £5.50 **Format** Cassette **Supplier** Imagine, Studio Sting, Mason Buildings, Exchange St East, Liverpool, Merseyside.*

CAN'T YOU just picture the scene at a certain office in Liverpool, some months ago?

"What shall we call this program of John's? It's about some troops making a dash across the countryside in an attempt to capture the enemy's port and headquarters. H h how about Zip-across? No-o-o, I suppose not. I know Z-Z-yomp! Still missing something, isn't it? P-S-S-tomp?"

We know it now as Stonkers, which is as daft a name as the others that Imagine dream up. However, the game is as good as they usually produce.

The two protagonists face each other across a river — a small-scale map is presented to the player at the start, showing the river, which opens out to the sea in the north-west corner.

This map takes up the top two-thirds of the screen, with a message area beneath. On either side of the map is a status board, showing the current strength of the computer and the player in combat and supply units.

The enemy port is to the left of the map, with the human player's port to the right, about mid-screen. His headquarters are in the top right of the screen, with the enemy's diagonally opposite in bottom left. In between are a couple of mountain ranges, and the all-important river, with (as in all good war simulations) just one bridge.

The rather simplistic aim for the player is to capture both the headquarters and the port of the enemy.

To move your man, Imagine have come up with a stunningly simple device: a large cursor which may be moved (under keyboard or joystick direction) over the unit you wish to move.

The area covered by the cursor may be magnified. On this large-scale portion of the map is a small cross-hair. Move this to one of the units represented by a little man (infantry), a gun (artillery), tanks (the armoured divisions) and trucks (supply units), and information on that unit is given.

Strength, morale and supply levels are detailed, and these parameters will determine how you use that unit. If you wish to move it, pick it up by pressing the appropriate key, or the joystick fire button, and move the cross-hair to where you wish the unit to end up.

Pressing the key or button again at the destination point will start the unit moving towards that point. Now you can go off and look at another unit, while the previous one is moving. Each unit has a certain range, though, and this, together with the varying terrain, affects the movement of the units.

During the deployment and the combat stages, each of your units must be kept properly supplied. This is achieved by ensuring that one of the four supply units is moved adjacent to the fighting unit you wish to "top up". Every so often a ship docks at your port, and vital supplies are channelled through the trucks, which are then spirited back to port to await orders.

After a certain amount of time (no, I won't tell you how long), the enemy units, having lulled the player into a sense of security, start to ▷

◁ move. You elect to see them or let yourself in for a surprise. When two opposing units meet, battle is joined and the victor is decided (I think) by comparing morale, strength and level of supply of each unit. The defeated unit is simply removed from the game.

There aren't really any more rules to Stonkers. The few that are included on the inlay should be read thoroughly. In one game I sacrificed all my units, but one, and made my objective of the enemy's headquarters wondering why the victory message hadn't appeared. I read the rules again, and of course, found that I should also have occupied the port.

I found it hard to be in two places with one unit, and had to admit defeat. There are some tactics that you will have to pick up as you play (even Monty didn't know all there was to know about the enemy).

Stonkers is a fairly simple sort of game; an ideal introduction to computer wargaming. It is to the great Eastern Front as Kriegspeil, for example, is to one of the huge Avalon Hill tournament board games; an ideal introduction. It may be simple but because of that, capable of becoming an exceedingly interesting way of spending the odd bit of spare time.

This is one of the advantages of Stonkers. A game should only last 15 or 20 minutes, so boredom won't set in. There are no large tomes of rules to sift through, no quick reference cards to get dog-eared and lost.

Colour is implemented well, and ticker-tape messages, written in a nice "World War II battered armour" script, chatter across mid-screen, informing the player of units in need of supply, units lost in còmbat and the amount of time the player has been "stonking".

The map is well-drawn, with contours, woods and marshes — apparently the map, in the design stage, covered more than 9,000 postcards, or something. It would be nice to think that Imagine could supply update modules, like Red Shift's, so that other campaigns could be fought, but the work is obviously daunting. There is a hint, however, that the same map may be used in the future with different scenarios which would be loaded into the master program (Mangle the Centurion or Nuke the Pike-Bearers, can't wait).

All in all, a great little program, and great value at the usual Imagine price of £5.50. Oh, the gnashing and wailing when that figure has to be increased in the future.

"Good report, Carruthers. Now I have a rather, ahem, sensitive job for you. We need someone to infiltrate this Imagine place and come back with the vital information; what Stonkers means. Good show, cut along now, old boy. Oh, and Carruthers, don't come back!" **TB**

A plot for the eccentric

Adventure *The Cricklewood Incident* **Micro** *Dragon 32* **Price** *£7.95* **Format** *Cassette* **Supplier** *Salamander Software, 17 Norfolk Rd, Brighton, East Sussex.*



ONCE UPON a time there was a dragon, guarding a heap of gold . . . that was a long time ago, though. These days dragons are no longer obligatory in adventures, and herds of Australians in Earls Court, Hell's Angels and alien spaceships, all of which are featured in this game, seem to be nearly as common.

Most adventures with eccentric themes like this do seem to include graphics, though, and this one doesn't — at least, I didn't come across any.

You are told at the start of the game that you are Arnold Q. Vole-Strangler. You are then asked to choose your character: Utter Wally, John Travolta, Genghis Khan, Superman or Geoff Boycott. Obviously Arnold is a skilled impressionist.

The game starts in a padded cell, from which you make your way onto a tree-lined road. There doesn't seem to be any obvious exit from this road, this being one of the annoying adventures where compass directions don't make sense (keep going North and before long you'll be back where you started). But not to worry, an alien spaceship will soon seize you and dump you somewhere more interesting. Or you may be picked up by the police for vagrancy, and simiarly dumped.

The first snag is that you need more money than you are initially provided with, to purchase useful items, bus and train tickets, and there doesn't seem to be an easy way of getting hold of any.

Perhaps my powers of lateral thinking just aren't up to it, but I tried everything I could think of, however silly, nothing worked. I didn't, ▷

Variation in war game scenarios

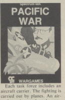
War game *Pacific War* **Micro** *Spectrum 48K* **Price** *£6.00* **Format** *Cassette* **Supplier** *Cases Computer Simulations, 14 Langton Way, London.*

I SUSPECT that whereas the attraction of adventuring lies in exploring new worlds, the lure of war gaming is the opportunity to change our own history.

In this naval simulation you take the place of the American fleet in the Pacific while the computer plays your Japanese opponent.

The setting is the Solomon Islands, whose green outline on a blue background forms a very attractive playing field. Pictured on the screen, as well as the islands, are the three American task-forces (represented by flashing numerals) and the American Henderson airfield.

At the beginning of the game you are shown three extensive areas of ocean in which the six Japanese task forces may be lurking, and in which you may accordingly position your own forces.



Each task force includes an aircraft carrier. The fighting is carried out by planes. An aircraft carrier can launch up to three missions concurrently (as can the airfield).

The purpose, target and complement (chosen from available fighters, torpedo planes and dive-bombers) is up to you. When the planes are despatched, charming little UDGs mark their progress.

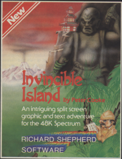
Since only the US fleet appears constantly on the screen, your first task is to find the enemy. When one of your planes flies over a Japanese ship, its location will be revealed and you can attack. Thus far the game is remarkably similar to that old favourite, Battleships. However, in Pacific War everything is in a state of fluidity so the ship you found may be lost again by the time you can get an attack squadron to it.

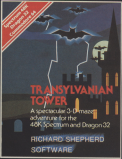
Although I didn't have the full instructions, which the game will include in the shops, the clear and sensible prompts, as well as simple menu selection made this a thoroughly enjoyable game to play. With nine skill levels and new starting positions for the enemy each time, I'm sure it will retain its compulsion and challenge for a long time.

It's a shame CCS couldn't have made it possible to play more than one game without reloading, but with each battle taking more than an hour, that's only a minor drawback. Overall I would call this a thoroughly praiseworthy version of a theme all too easy to make boring. **DD**

RICHARD SHEPHERD SOFTWARE

ELM HOUSE, 23-25 ELMSHOTT LANE, CIPPENHAM, SLOUGH, BERKS.
CREDIT CARD HOTLINE 06286 63531





Invincible Island by Peter Cooke

Exhaust your ingenuity in the quest to find The Seven Parchments of Xaro and their meaning! Will they lead you to undreamt of treasures or eternal doom? Explore the mysteries of the stockade and puzzle within the Pagoda! A spectacular split screen graphic and text adventure to braintease you for weeks! Adventurous graphics for every location. Save routine. **£6.50.**

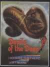
Transylvanian Tower

A spine chilling adventure . . . enter via the dungeons . . . navigate your way through 500 3-D rooms . . . survive the swooping vampire bats . . . reach the terrifying top . . . confront and kill Count Kreepie, ridding the world of this Transylvanian Terror.
Can you survive the Top of the Tower? **£6.50.**

SELECTED TITLES AVAILABLE FROM

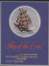
W.H. SMITH, JOHN MENZIES, BOOTS AND ALL LEADING COMPUTER STORES

"ADVENTURES INTO IMAGINATION"











NEW DEALERS WELCOME — RING (06286) 63531 FOR DETAILS

therefore, make much progress with the game.

The other initial problem is the Hell's Angel who is intent on beating you up. You are offered a choice of tactics to try against him: poking him in the eye, strangling him, kneeing him and so on — but even if you succeed in fighting him off once, he reappears in next to no time ready for another fight.

I didn't get far enough into this adventure to say categorically that it's no good — if you have a sufficiently perverted sense of humour you might even enjoy it — but it's not to my taste, thanks. **MN**

Still a classic braintease

Adventure *Classic Adventure* **Micro** *Spectrum 48K* **Format** *Cassette* **Price** *£6.95* **Supplier** *Melbourne House, 131 Trafalgar Rd, Greenwich.*

BACK IN the dim, distant past (well, 1978 actually) two bored mainframe programmers came up with the idea for a game. They used over 200K on their mainframe to write it in Fortran, and when they'd finished they called it Adventure. Six years and several computer lifetimes later the Classic Adventure is available for the Spectrum 48K.

Actually, Abersoft have been selling the same adventure for some time, and there is a first-class version from Level 9 under the name Colossal Adventure, but Melbourne House's commercial muscle will doubtless win many new admirers for this classic textual brainteaser.

The screen display is an elegant white-on-black, very restful on the eyes when the first location pops up on screen and you find yourself on a deserted road beside a building. The plot is familiar (but bear in mind it is the original), sending you into a



labrynthine cave system in search of treasure, and inviting you to plunge in and explore.

Respect for its pivotal role in adventuring history would not alone make this a game worth playing. However Classic Adventure has much more going for it than mere historical significance. The descriptions of the locations are detailed, literate and frequently lengthy, while the locations themselves are varied, unusual and fascinating. Starting out is a pleasure, since there is plenty to explore at the beginning before the problems really being.

The problems are thorny and unpredicatable — some have very logical solutions, while others defeated all my best efforts until in a fit of pique I typed something quite daft, only to find that was what I should have done all along. Crowther and Woods, the mainframe programmers, have a wicked sense of humour, but they are always fair.

The caverns house a selection of flora and fauna ranging from fluttering birds, fierce snakes and baleful dragons to axe-wielding dwarves and larcenous trolls. These are considerably less agressive than their Level 9 counterparts, which is a relief, and Classic Adventure is refreshingly calm, allowing the adventurer to get on with exploring without the constant danger of attack.

On the minus side, some of the descriptions are mis-spaced or mis-spelled, the vocabulary, while extensive, is a little picky and some of the responses can be odd, but these are very minor complaints about an excellent games in the future. **DD**

Puzzling plot for player

Adventure *Solaris* **Micro** *Spectrum 48K* **Format** *Cassette* **Supplier** *Softel, 5 Durward Drive, Glenrothes, Fife, Scotland.*

THE cassette cover boasted: "Full colour 3D high raised graphics". I had a bad feeling, any firm who cannot spell on their cassette covers are obviously not taking the care necessary.

The game has a very strange plot: you are going to be taken to a space ship upon which you will travel to a prison planet. You have to stow away on the ship, leave the ship early to go to the planet, you have to find the encampment in which you are to be put, and lead the prisoners to rebellion. Surely it would be easier to let them take you there and just control the rebellion.

The game is in three parts, all basic programs. The first part is the instructions, the second is the adventure on board the starship, and the third is, I presume, the adventure on the planet. I say presume because this is one of those adventures that hasn't been finished and never will be.

The game is boring, and impossible. Softel sent a hint sheet with my copy, though I doubt it will be widely distributed. I progressed using the hints on the sheet and then my progress stopped.

The first location is the ground car. The drawing shows two boxes in a room with a door at the end. There is only one exit. The door is locked, and you have nothing with you. The room has nothing in it. Sounds like a good meaty start doesn't it? Unfortunately you would have to be a supernatural-lateral thinker if you want to solve the problem.

The solution, for those who have bought this game and are totally stuck at the first location, is to LIFT THE SEAT (of the ground car, whose seats are not shown in the picture), TAKE THE EXTINGUISHER (which is under the seat), PULL THE PIN (which for some inexplicable reason does not set the extinguisher off) and PICK THE LOCK WITH THE PIN.

The hint sheet says that the puzzles get more difficult as the game progresses, is this possible?

The programmer gives a warning that pressing break will cause an immediate system crash. This is a simple lie obviously designed to put off the pirate/cheat.

The game cannot really be recommended to anyone who does not have an IQ of 350 and a mind like Dr de Bono. If the sound of the puzzle above interests you, and you think that you could have worked out the solution with no clues, then buy the game, otherwise, steer clear, and spend your money on something else that is perhaps, dare I say, easier. **JO**

A star among strategies

Strategy *Star Trader* **Micro** *BBC B* **Price** *£7.95* **Format** *Cassette* **Supplier** *First Byte Software, 10 Main Centre, Derby.*

PERHAPS the best place to start is with a few grumbles about this game. The cassette cover is a wonderfully produced glossy affair. The downfall is the lack of any other information, such as the type of game, the scenario or a brief outline of the instructions. One would expect this information as standard, especially for almost £8.00. If this will be remedied soon the game would be good value for money.

The game is loaded with the *RUN command and I found no loading problems. The only drawback here is it won't work on the Electron. There is a fairly long wait before the game starts, but it is worth it. The player now encounters the instructions and discovers the purpose of the game.

The player moves around a few different constellations looking at and exploring various planets. Should you try to explore a star or something equally as silly the game tells you of your

folly and continues. The screen is split in four. Moves are in the top left and three other screens are used for information. The rules and prompts are displayed in the middle of the screen and scrolled up.

Apart from surviving, the player has to find planets that are friendly, inhabitable and occupied. When a plant that fits this description is encountered trading may take place. You will need basics such as food, water and fuel. You may also be able to trade for minerals. The overall point is to be in a strong enough bargaining position to buy the master computer.

To buy the master computer you must have collected enough of the right kind of goodies and then you can save the world and finish the game. The player can view each planet and is given a range of information for individual planets.

Should the player then land on a planet more information will be displayed, such as the minerals and basics available for trade.

It is of course wise to check out that there is enough oxygen on the planet. The game is not easy to solve and poses some interesting problems for the player. Beware of landing on or hanging around hostile planets, or you will find yourself exterminated.

The overall conclusion must be that Star Trader makes most other strategy games look like non-starters. The screen display is well presented and legible. The response time could be a little quicker, but it does not hamper play. The lack of plot and instruction on the cassette cover is the main drawback. Providing this is rectified, Star Trader is good entertainment.

First Byte say that Star Trader will be available for the Electron and that its release is imminent as is Valley of the Pharoahs. They also say that there will be an entirely new range of software for the Electron and that it will be of the highest quality. We shall have to wait and see. If Star Trader is used as a yardstick, we can expect some good games in the future. **KB**

Treasure hunt mystery

Simulation *Micro Driving* **Micro** *Spectrum 48K* **Price** *£6.95* **Format** *Cassette* **Supplier** *Softel, 5 Durward Drive, Glenrothes, Fife.*

WHEN you're tired of slogging it on foot around tunnel, cave and forest maybe you'd like to relax with a pleasant drive around an island resort?

You could even get involved in the annual treasure hunt, which, like the popular real time (and real car) game involves following a series of clues to a series of objects until you've found all seven when you win the game.

Unlike adventures, you do not know what you are looking for in advance and you may not know it even when you've found it. However, despite its contemporary setting, it is a traditional collect the mysterious objects kind of adventure, and not much different from many except in the kinds of objects available. For example you start with some pounds, some dollars, a striped tie and a piece of wire. Not much dungeon aids.

All the locations are illustrated with Hobbit-like graphics, which threaten to become standard on this kind of game. They are almost as good as The Hobbit's and are drawn much more quickly, but are generally of full objects, such as walls and gates. I cannot see the attraction of such graphics in a game, but many people like them.

However, some of the graphics are good, especially the views of the road and dashboard as you drive around. While not as convincing as a real-time simulation, the driving part of the game is quite satisfactory. And would be more so if you were allowed to do everything that a driver could do. But you cannot reverse and sometimes, cannot even get out of the car.

Some of the error messages in this game are more irritating than most. This might be because elementary instructions such as OPEN GATE or CLIMB GATE just meet obstinate refusal but it's also because the tone of the messages veers from sarcasm to downright insult.

The most irritating part of the game is its input routine. I think the design for typing in instructions is a major programming error. Instead of the normal Spectrum INPUT prompt a custom made flashing prompt occurs. It refuses to go away and fails to collect about a third of the input typed at normal two finger speed. Anyone who types their instructions quickly will find themselves cursing again and again as commands are ▷

Taking charge of air traffic

Simulation *Air Traffic Control* **Micro** *BBC B* **Price** *£8.00* **Format** *Cassette, Disk* **Supplier** *Microdeal, 41 Truro Rd, St. Austell, Cornwall.*

THE MANUAL which accompanies this program describes it as "a computer model of an air traffic control situation".

Unlike an ordinary flight simulator which gives you control of an aircraft, you take the position of the air traffic controller attempting to fly up to five planes at once.

Points are scored for successful landings and subsequent takeoff. Each plane can be landed as many times as possible, and a high score will enter you in the Hall of Fame.

At the beginning of the simulation the screen shows various dials and meters with a runway and initial five aircraft. They have random altitudes, velocities and directions so it is best to begin by checking each plane and setting it on a safe course.

A plane can be lost by leaving the screen or hitting the ground at this point. I preferred to kill a few of them first, as the screen updating is very slow with all five in the air.

To check a particular plane, the fire button or space key is



pressed. This freezes the action and a correction can be made, when a cross appears and must be placed over the nose of one of the planes.

If the chosen plane is in the right place and altitude the program immediately moves on to the landing stage, otherwise the necessary corrections can be made and the program carries on.

For the landing situation, the screen clears to show a horizon line the runway and a rear view of the plane. You now have direct control over the landing of the plane, and 10,000 feet of runways to get it right. The exact position above the runway depends on the position the plane was at when it was last checked, so that each landing approach is different.

To land successfully, the plane's altitude must be reduced to zero with a velocity of around 225 feet a second.

Points are awarded according to how close the touchdown was to the ideal. Although it is not mentioned in the manual, you can come down perfectly but crash because the angle to the runway was too steep.

After a plane has been landed the program returns to the original screen with the landed plane shown at the bottom of the picture. If it crashed the plane is taken out and the program returns as normal. A plane which has landed can of course be taken straight up again.

The cross is positioned over a landed plane in the usual way, and the display clears to the horizon line, runway and plane rear again.

To take off you merely press fire to fire up the engines and pull back the stick when your velocity is over 200 feet a second.

This program certainly is a realistic simulation and will please those who want more than a simple simulator. **MW**


REBELSTAR
NEW
RED SHIFT
REBELSTAR RAIDERS
REBELSTAR RAIDERS
. . . He manipulated some keys and the "MOONBASE" defences "Blue Print" filled the screen . . . The mutant clones and that monstrous organic computer "MAIN-COMP" really did a perfect job . . . "STARLINGALE" and "FREEDREAM" were all be had! Now or never, Jo Capricorn thought . . . He punched the starsystems co-ordinates in the navigational computer and in a silent "Thud" the stars flickered past him as the starship "STARLINGALE" jumped in to light speed. The Battle had begun . . .
Superb Graphics — Amazing Detail
Descriptive Sound . . .
SPECTRUM 48K
£9.95
APOCALYPSE
. . . Some hard negotiations had to be made before unleashing the mighty armies — The 4 maps on the screen looked convincingly deceptive — Nuclear war could break out any minute
For SPECTRUM 48K — BBC MODEL B
£9.95
Expansion kits for Spectrum 48K
(Ask for other options available)
(Microdrive version available by
RED SHIFT
Apocalypse
REDSHIFT LTD
12C MANOR ROAD
STOKE NEWINGTON
LONDON
N16
TEL: (01) 800 1333
EMS
Apocalypse is produced under licence from the Games Workshop Ltd.


R RAIDERS LAUNCH...
NEBULA
nebula
A Game of Galactic Strategy
THE NEXT WINNER FROM REDSHIFT
Time Lords?
First Empire?
IN THE MAKING!
*Distributors, Multiple Store and Retail Shop inquiries invited.
HOW TO ORDER
You may purchase any of the items listed by cheque/P.O. All you have to do is fill in the details in the coupon enclosing your cheque/P.O. made payable to BINDERMAN LTD and your order will be despatched within 7 to 14 days. All prices inclusive of 15% VAT and P & P.
OVERSEAS ORDERS — Please add £0.50 per item ordered.
Post to; REDSHIFT LTD, 12C MANOR ROAD, STOKE NEWINGTON, LONDON N16.
Please send me the following items; (tick item required)
REBELSTAR RAIDERS (No. Req...................)
NEBULA (No. Req...................)
APOCALYPSE (No. Req...................)
Please send details on Apocalypse and other games.
I enclose my cheque/P.O. for £...............................
NAME ....
ADDRESS...............................
TEL: (Day) .... TEL: (Eve)...............................
Please allow 7 to 28 days for details on Apocalypse and other games.

◁ misunderstood by this excruciating procedure.

On the whole this is a competent game let down by one or two annoying mistakes. It strikes me as a game designed by a programmer rather than a games player. The user has not seriously been thought about, which is a shame because if you can get over the irritation it looks like anb absorbing adventure. **NW**

Ten Indians ambush Spectrum

Adventure *Ten Little Indians* **Micro** *Spectrum 48K* **Price** *£9.95* **Format** *Cassette* **Supplier** *Digital Fantasia Ltd, 24 Norbreck Rd, Norbreck, Blackpool.*

MANY OF you will be familiar with this Mysterious Adventure, as it has previously been released for other popular micros, but this is its first time out for the Spectrum.

The scenario is based on the death of one Major Johnstone-Smythe who, to stop any of his money-grabbing relatives inheriting his fortune, had all his assets converted to gold and cast into a figurine which he then hid.

He also had 10, worthless figurines made from various materials and hidden in and around his mansion. A nephew informed the World of the Major's activities and it is generally accepted that one cannot obtain the treasure without 10 Little Indians. You, needless to say, are going to attempt to recover it, although many others of a similar ilk have never been seen again.

I have to make the point that this adventure is definitely not designed for beginners. The tricks and traps start at location one and appear with alarming regularity thereafter. I have, so far, encountered four death-traps, two extemely complicated mazes and several apparent dead ends which turned out to be anything but.

After five hours work on the adventure I have found only two Indians and some to yet another dead-end. I know there are more locations to explore but I can't find them. No-one should expect quick results.

I hate waiting for a picture to form at each location before I can continue so I was pleasantly surprised to discover that one can switch to text-only simply by pressing the enter-key. The graphic displays are good and the text is concisely informative. Inptus are of the normal variety and a save-game feature is included.

The packaging is sensible and well presented and includes the offer of a help-sheet for those, like myself, who feel they may need it.

I think I can safely say that Ten Little Indians is value for money and should attract just as many addicts as any other quality adventure. **SC**

Prepare for a surprise

Adventure *Kingdom of Hamil* **Micro** *BBC B* **Price** *£9.95* **Format** *Cassette* **Supplier** *Acornsoft Ltd, 4A Market Hill, Cambridge.*

"YOU ARE the rightful heir to the Kingdom of Hamil, but, stolen from your royal parents as a child, you have only just discovered your birth right." So starts the latest Acornsoft adventure.

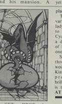
The adventure starts with your seeking sanctuary in a small chapel. Your aim? Prove your identity, thus regaining your rightful inheritance. To do this you must collect all the treasures and deposit them in the vaults of Castle Hamil. Then you are asked a question which you must endeavour to answer.

This program has the feel of a standard adventure, but there are none of the dwarfs or dragons to be disposed of. If you stay in one place for too long, a giant weasel comes along to hasten your demise. This is quite a surprise at first.

At one point along the way a vampire has to be disposed of in a clever way. (Not the same way as in Sphinx.) The ever-present maze is very simple, which is a boon for the less experienced player. However, the Lost World is very difficult to sort out.

The descriptions are the same everywhere, and to make matters worse, there is a Tyrannosaurus Rex after your blood, preventing you from re-tracing your steps. Naming this area the Lost World was very appropriate, even making a map of it I am totally lost.

The program lets you carry a generous eight items. A time-saving feature is that by



entering GET or DROP only, the first item listed is collected or dropped. Abbreviations are accepted to four letters. The response time of this program is not really that long, but long enough to make it irritating.

A full description of each room is given only the firs time you enter it. The full description is supplied when one LOOK's around. The shorter descriptions can give helpful hints as to the use of a room. I would have thought that the memory could have been used to better effect than to hold two sets of descriptions.

Kingdom of Hamil would probably be most appreciated by the moderately experienced adventurer. It is, in some places, very easy, and in some places very hard to play.

The creatures encountered vary from a sweet little old lady who wants you to come to dinner, to a lonely baby Cyclops desperately in need of a dinner. A little knowledge of history comes in handy at one point.

Some more humour throughout the adventure would have been a good idea. Kingdom of Hamil is a good program, though not brilliant. My main complaint is that it lacks imagination. **AT**

More slicks than kicks

Strategy *Gusher* **Micro** *Commodore 64* **Price** *£9.95* **Format** *Cassette* **Supplier** *Visions Software, 1 Felgate Mews, Studland, London.*

THE AIM of the game is to collect oil from well heads and to transport it by means of lorries and/or pipelines to ports where it is loaded on to tankers and super tankers for export.

When the oil is exported, it earns the player money, and the actual aim of the game is to earn £1 million.

The game is played on a map showing the well heads, refineries and pipelines. The map also shows the movements of the oil by means of the ships.

In my opinion the game is designed to bore the two to four players it was written for.

Each player is given a turn in sequence. At this time, the game will generate such things as "Oil at well head", "move oil by pipeline" and will then allow the player to decide what to do with the current holding of oil: whether to export some, or to transport more from the well head to the refinery.

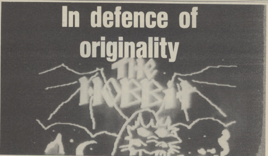
It is the sequence of random events that drives the game along. Also introduced are various hazards which make more difficult the handling of the oil ("pipeline burst"). The introduction of these hazards provides the element of chance within the game.

So the playing of the game involves little more than sitting around waiting for the game to inform you of the next random event that has been generated.

One fortunate feature of the game is its many sound effects — without them, most players would probably fall asleep. This is a deliberately boring review. It is meant to give you some indication of the joys to be encountered when playing Gusher. **RJ** □


Fancy a game of dragon slaying and dungeon looting? All the best adventures can be found in
THE ROLE-PLAYING GAMES MONTHLY
WHITE DWARF
White Dwarf is the monthly magazine of adventure gaming. Dungeons & Dragons, RuneQuest, Traveller, computer games, comic strips, news and reviews await you each issue, and fantastic colour artwork too.
White Dwarf is available on the fourth Friday of each month at your news-agent or games shop. Order your copy today. Only 85p.
The Giant in Fantasy

In defence of originality



Laurence Miller on maintaining standards in adventure software

ADVENTURE AND adventures are familiar terms to the majority of computer users from the mighty mainframe operators to the rather larger group of micro users. Unfortunately there exists considerable confusion as to what these terms actually describe and as to the origin of such programs.

The first adventure (sometimes called just that), Colossal Cave, was written in the mid-70's by two Americans, Crowther and Woods, and is said to be an attempt to computerise the role-playing game of Dungeons and Dragons (D&D) which was released in 1974 by TSR, Inc.

Whether or not this is the case doesn't really matter as any serious D&D player knows that Colossal Cave bears about as much resemblance to D&D as Monopoly has to the way a government runs the country.

The majority of adventures are both played and designed for playing with a single character rather than multi-player as D&D and the majority of other role-playing games (RPG).

Analogy

The closest analogy is a game called Tunnels & Trolls which can be played as other RPG's but, like most adventures, is usually played solitaire with the aid of pre-programmed adventure books rather than computer programs.

This is not a put-down of adventures, more a plea that when people make comparisons they should compare like with like and be certain that all concerned know what is being discussed.

When the first adventures were written (Colossal Cave, Zork), the programmers were professional, widely read (especially SF and fantasy) and had vivid imaginations. They were writing as experimentation in the field of artificial intelligence research and/or for their own and friends' amusement rather than for commercial reasons.

The result of this is clearly visible in that Colossal Cave and Zork are still with us and recognised as classics even when running on their micro-computer brethren, despite the plethora of text and graphic adventures now available.

No-one seriously thought of trying to put a program of this type onto "one of those micro-toys" until a professional programmer called Scott Adams got hooked on Colossal Cave and decided, much to the amusement of his friends, to write a similar program on his TRS80 Model 1 with 16K of memory.

It wasn't easy but eventually he succeeded and subsequently produced 11 more adventures (10 if you count Savage Island I and II as one game) all of which are still with us.

From the same type of background as Colossal and Zork, these programs are still popular, so much so that they're now becoming available on UK micros such as the BBC and Spectrum . . . keep your eyes open for Number 13 due out soon.

To some degree this type of programmer is still with us if you look at adventures such as Valhalla where the main programming belongs to professional mainframe programmers but with the advent of large scale general ownership of micros a new breed of programmer is with us bringing both good and bad results.

You do not need to be a professional mainframe programmers, but with the adventures, this is not the main point of the preceding paragraph. The real point lies in ". . . widely read (especially SF and fantasy) and had vivid imaginations", something which seems to be sadly lacking in the current spate of adventures starting to flood the market.

Part of the cause lies in a recently discovered profitable line of books concerned with *How to write Adventures on Brand X computer*, which provide valuable technical assistance with the nuts and bolts of adventure writing but don't (and cannot) provide that creative spark, originality, whatever, which distinguishes a great or even good adventure from the ravening hordes of mediocre and downright bad adventures.

Brilliant

Contributing to this we now have programs such as The Quill, which are technically brilliant. In the hands of the truly imaginative and creative person such programs can remove much of the drudgery from writing, allowing more effort to be expended in the act of creation but more likely to encourage the sausage-factory effect. Combine all this with the commercial factor and the net effect on a potentially superb, new form of entertainment could be disastrous.

The commercial effect involves many things, such as the discovery that "there's money to be made in these adventures, all you need is some caves, a few monsters, or a spaceship, some treasures and a few deaths".

Then there is the realisation that text adventures don't attract arcade gaming addicts so easily so "let's push ▷

◁ graphic adventures''. Now pictures add to adventures but it should be kept in mind that the best pictures often come from the players' imaginations. Try playing The Hobbit on the BBC — it's the same game as the Spectrum and Commodore 64 but it doesn't have pictures — you don't lose any information necessary to solve the game as it's all provided in the text and you're free to visualise Gandalf, Thorin and all the others according to your own view rather than someone else's view. The game moves faster without the pauses required and each picture is drawn (and that does allow for BBC operation being generally faster).

All of this could pale into insignificance when compared to one overwhelming fact. A large number of programmers who get their work published are dedicated micro-computer users whose main hobby is computing. They have to be to produce programs with a physical quality suitable for publication but this also means that their experience in other areas is limited — again by definition since they devote most of their spare time to computing.

This shows in the types of programs appearing in large quantities both in arcade games and adventures. So many lack originality, spontaneity, even elementary logic. As an example, without mentioning names I recently tried an adventure which exhibited all of these points.

You start play on the bridge of the Enterprise (why copy someone else's plot, you could always provide a few notes about your own space epic as background). All rooms leading off the bridge, bar one, contain nothing you can use at that time and nothing you can pick up.

The exception is totally dark when you enter (the infamous 'totally dark' room of a hundred other adventures) and if you move, you die (remember that there was nothing to pick up elsewhere and you are not carrying a lamp).

Fruitless

After fruitlessly searching for a lamp and dying several times, in sheer frustration I entered ''Light Lamp'' despite not having one and lo there was light. Was a deadly precipice revealed? An electrified wall? An automatic laser? No such luck. I was simply standing in an elevator with its controls visible.

If anyone has a logical explanation as to how you die by moving in an elevator rather than just bumping into a wall let me know. This is a basic example of an unoriginal, illogical and unfair program. Surely it would have been much better to bump into the wall and then to have fumbled about for a light switch with a good chance of hitting the elevator controls by accident and thus startling me with a message saying, ''the door has closed behind you and the room is moving'' or something similar.

Unoriginal ideas such as the proceding one should be of concern to all adventurers and software houses because the ultimate decisions, relating to the quality of the game you play, rest in your hands. If software buyers are not discerning the quality of originality of the products would sink to the level of the lowest common denominator and discourage good writers from the necessary effort. Why spend months writing a brilliantly original adventure when the return can be the same from a rip-off hack job.

The other side of the coin rests with the software houses and whether they have the integrity to retain the necessary quality in their products or go for the cheaper option of trying to market as much product as possible regardless of quality and originality.



Almost anybody, who can program a little, can write an adventure but it takes a lot more effort to produce a good adventure, which would you rather see happening?

To help in your decision perhaps a few of the ingredients, which go to making a good adventure, should be considered. The prime necessity is a good storyline with the adventure being akin to a novel which you don't just read but play an integral part.

There should be a reason for the game whether it's to get rich, solve a murder, save the world, destroy an evil wizard or whatever and the program should maintain its internal logic to be consistent with this reason.

This doesn't rule out a considerable number of sidetracks such as the necessity to become rich and powerful before taking on the wizard or leaving piles of treasure around to distract the unwary from their quest.

It isn't even necessary to specify the reasons before starting an adventure, as it can be fascinating to discover your real purposes as you explore the programmer's world but there should be a reason.

If is very frustrating to be killed for no good reason and a good adventure will always give some warning of impending doom or a way out, no matter how obscure.

Puzzles and problems are the meat and drink of adventuring but shouldn't involve knowledge of a too specialised nature for their solution especially if such a solution is crucial to survive and/or win.

For instance it's neither fair nor useful to tie a vital clue to an esoteric piece of knowledge about the Cray 1 computer even if the author uses it every day — there aren't that many of us with access to a Cray.

Graphics are unnecessary to adventures, after all Zork is one of the best selling programs for personal computers in the US and not a picture in sight. But that doesn't mean they can't play an important role. At the moment graphics have been discovered by adventure producers, especially after the success of The Hobbit and Valhalla with everyone jumping on the bandwagon. Part of the problem is that in the UK the use of disk drives is far lower than in the US where the graphic-assisted adventure is reaching new heights by using multiple loading from disk.

Here, with the majority using cassette loading, it is necessary for the entire program to be memory resident, which means graphcis take a disproportionate amount of memory compared to their value in the program as pictures eat a lot of memory. The net result is the release of a few adventures where the graphics are either good or of genuine value to the program and an increasing number in which relatively poor graphics are being used as an excuse to cover up even poorer adventures.

There are also available a new breed of games which rely heavily on graphics, the so-called arcade-adventures (or even arcventures) such as Halls of The Things (Crystal), Atic Atac (Ultimate) and Warlock of Firetop Mountain (Penguin), which are proving very popular due to their originality. So popular are they that they are spawning their imitators, who are already lowering the tone of the neighbourhood.

Sausages

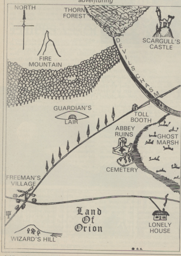
The overall plea to programmers must be, ''Please don't turn out sausages,'' because, sooner or later, they'll learn they can't be fooled indefinitely.

If you turn a chance at originality into mediocrity at best and boring, repetitious tedium at worst then no-one will buy your work. Before starting to code your program ask yourself some questions. Is my idea original or does it hve a different slant on an old idea? Am I giving everyone a fair chance (if they think about my problems)? Do the graphcis add to the game (and is the quality worth it) or would the memory be better used in expanding the text?

As an alternative, try working in a team effort — after all Colossal Cave, Zork, The Hobbit and Valhalla all involved two or more people working together, even Scott Adams no longer writes an entire adventure on his own. He uses a team approach to final production and artists to produce the graphics. Remember that in the final exercise, putting a program together is only coding but writing an adventure is primarily writing an interactive novel. □

Meeting the programming challenge of Orion's Land

***David Newton** finds a way of ensuring variety in adventuring*



HAVING PLAYED and conquered some dozen or so adventures in my 12 months of computer ownership I began to have feelings of *deja vu*. It was all becoming too familiar and the challenge was waning.

The idea of writing my own adventure occurred to me and in August 1983 I started work on what was to become The House of Orion for the ZX Spectrum.

On the face of it, this was an ambitious project for me as my previous experience amounted to one character generator for a certain role playing game. Prior to that I had hardly looked at a program let alone written one.

The generator taught me several invaluable lessons though, not the least important was that if I wanted to write a program I could.

An adventure game, however, is an entirely different proposition and the problems I encountered were new to me. For instance, where do you start?

Forget the machine for a while and think plot. I drew maps, doodled and thought and out of these fragile beginnings my inspiration was born. The markings on one of the maps reminded me of something. I dug out my copy of *The Observer's Book of Astronomy* and there was the constellation of Orion.

Patrick Moore had included in the text references to the mythology and legends surrounding Orion the hunter and my plot slowly drew together. The adventure was to be set in the land of Orion at a time long ago when magic was still a power, when the constellations had not yet been set in the skys and the heroes and beasts that are the constellations still roamed the Earth. The various stars in the system became the characters with Rigel the King and Bellatrix the compulsory Wizard.

There had to be a villain of course and Scargull was created, the master villain who flung down the royal house of Orion.

Object

This was to be the object of the game then, to destroy Scargull and restore Rigel to the throne. It's a great deal more complicated than that but I think you'll get the general idea.

I again worked on the maps and this brought the land of Orion into existence. I included eight basic locations which were later divided into some 120 sub locations. This seemed to be about right but I kept it flexible and was able to expand or contract it as need arose. It looked like a good sized area for any hero to explore.

I then worked on the puzzles that adventurers expect and the ideas came thick and fast. Before I knew it I had a full scenario and was ready to return to the keyboard.

It was now apparent that the format of the game was to be the traditional text only. It was going to be too large for any form of graphic representation. This fitted in with my preferences though and

without graphics I could expand it to the complexity I was planning on.

In this type of game there are not that many different routines that you actually need. All you require is a method of describing the player's location, allowing she or he to input a command, decoding that command and printing the consequences of that command.

The heart of any adventure program is its catalogue of objects to be found and used, any text only game would be a dead affair without these. We need then some way of defining whether or not our player is in possession of any given object.

A simple and effective way of doing this is to assign a variable to each and every object in the game. You can do this in the first few lines of program in a long list of LET commands. If you give each object a value of 0 then they don't exist as far as the computer is aware and the player doesn't have them.

Similarly a door could be given the value 0 when it is shut then changed to 1 when opened. You can now test if a condition is met by using IF or IF NOT.

Orion starts with a list of 45 LETs and in it is the statement LET loc = 2000, loc in this case being short for location and the number 2000 being the line number of the starting location. A simple GOSUB loc will now take the player to the hill top and the start of his journey. (See listing one).

```
2000>LET l$="You are on the hill
side.There   are paths leading u
p and down.": LET up=1390: LET d
o=1420: RETURN
```

Listing one: movement to the hill-top and start of the journey

```
 102>FOR n=1 TO LEN i$: IF i$(n)
>="A" AND i$(n)<="Z" THEN LET i$
(n)=CHR$ (CODE i$(n)+32)
 103 NEXT n
```

Listing two: weeding out upper case letters

```
 501>IF i$(1)=" " AND LEN i$>1 T
HEN LET i$=i$(2 TO ): GO TO VAL
"501"
 502 LET st=0: LET v$="": LET n$
="": FOR n=1 TO LEN i$
 505 IF i$(n)=" " AND NOT st THE
N LET st=1: GO TO 520
 510 IF NOT st THEN LET v$=v$+i$
(n)
 515 IF st THEN LET n$=n$+i$(n)
 520 NEXT n
```

Listing three: changing into verb and noun strings

```
 272>IF v$="turn" THEN GO TO VAL
 "1980"

 1980>IF n$="" THEN INPUT "Turn w
hat?:"; LINE n$: GO TO VAL "1980
"
 1985 IF n$="handle" AND loc=1032
 THEN PRINT '"The door opens.":
LET we=1135: GO TO VAL "100"
 1987 GO TO VAL "499"
```

Listing four: examples of IF . . . THEN lines using recognised verbs

Underway

If L string is now printed the game is underway. You can see that two directions are possible from the hill top — up and down. If at the start of the main loop we put a line defining all directions as 0 and we now change two directions to a value greater than 0 then you may see that the player can only move in a direction that has a positive value.

In this case that value represents the line numbers of the adjacent locations. If after every change of location we reset all directions to 0 then we are ready for new paths to open up.

If the player chooses to go up the hill we can test if up is possible again using IF, redefine up as loc and move to the line number representing that direction.

Now we have to contend with the player's input. This may be a direction he wishes to go in, a desire to examine or take an object, a plea for help or indeed anything that comes into his head. Therefore, the program must be able to react to as many commands as possible and have a reaction prepared for invalid inputs. A large a vocabulary as possible is necessary then for the average adventurer, a creature with a permanent glazed expression and bruise marks up and down his body where he has kicked himself for not thinking of that before.

When the command is input it receives a series of transformations prior to the result being printed to screen. First the upper-case letters are weeded out (listing two). This two-line routine can save a lot of heartbreak when trying to debug the final program. Next the input must be split into the two-word format verb noun. Listing three changes the i (input) string into the v (verb) and n (noun) strings and from now on these two words are dealt with separately.

I constructed a list of IF . . . THEN lines using every verb recognised by the game and then diverted the verb through this list. Directions can also be accounted for in this list. For example IF v $ = "up" AND up (has a positive value) THEN LET loc = up:GOTO 25 (where up is reset to 0). Listing four shows another example. The verb turn has been input and this sends us to line 1980 where the references to turn are stored. Line 1980 checks to see if a noun is also present and then searches for that noun.

It finds that only one thing can be turned, a handle, and then checks to see if the player is at the same location as the handle. If he is, it informs us that the door opens and opens the route west. The program is then sent back to line 100 where the player is again asked what next.

Line 1987 here is interesting. If no references to the noun are found or a variable not fulfilled then line 499 is used to tell the adventurer "I can't ";v $" ";n $;" try again." A similar line just prior to 499 and at the end of the v string routine informs the player that his chosen verb is not understood. Thus this section of the program is error trapped.

Take and drop may cause you some problems. The take section of my program relies on the variables initiated at the start of the game. If the player indicates that he wishes to take an object and we confirm he is at the right location by checking loc then we change the relevant variable to the value of 1. If an inventory is now requested simply check the value of all variables and print to screen all those with a positive value.

I found that the use of DATA and READ were not as vital as some people would have you believe but without these I had a few uneasy moments with drop. I found though that I didn't really need a drop routine.

Most of the objects in the game were dual purpose, ie required in more than one place. So if an adventurer asks to drop an object he is asked "Why? You may need it." Occasionally I allowed a strange creature to sneak on screen and run away with a dropped object never to be seen again.

We now have a method of describing the player's location, decoding an input command and allowing the computer to analyse that command describing the consequences.

Skeleton

What more do you need for an adventure? Not a lot really. All the rest is ancillary to the main program. Once you have the main skeleton of the game laid out you can go through it again adding professional touches here and there. It's the bare bones of a program that make the game.

I hope this article will inspire some of you to attempt the seemingly quantum leap from playing games to compiling them. You'll find that the buzz you get from solving an adventure is nothing compared to actually writing one. I'll supply a cassette of The House of Orion to any readers who send me £3.00. (Write to David Newton, 4 Pewfist Green, Westhoughton, Bolton, Lancs). □

Putting code in the picture

*Software author **Stuart Sampson** looks at the advantages of writing adventures using machine code.*

PERHAPS THE most obvious use of machine code in adventures is to service graphics at speed.

"Graphics?, at speed?, it's not an arcade game, you know" might be the cry of the traditional adventurer, and I must to some extent agree, but graphics are used to illustrate adventures, and in some cases they are animated as well. However even for text adventures there is quite a lot to be gained from throwing off the fetters of BASIC and getting the machine to do exactly what you want, using its native tongue.

It rather depends what you want from an adventure. They span the full range from those of great size, having hundreds of locations, terse text and the accent on quantity, to full three-D graphics, sound and real-time animation, with the accent on realism. To fans of the former it is a bit like: you've read the book, now go and hate the film.

I shall introduce the ideas of machine code as applied to an adventure using traditional sentence input, hi-res colour graphics illustrations, and implemented on the Dragon 32. The main reason being the fact that I am writing just such a game, but I shall try to avoid being too specific, and keep the cat in the bag. So, non-Dragon readers, read on.

Polling routine

Unfortunately the Dragon has more than its fair share of restraints, and this is definitely a mixed blessing. You cannot use text on the hi-res screen and the input routine leaves a bit to be desired, so I have really got down to fundamentals. My adventure has its own keyboard polling routine and its own character generation, and can almost work independantly of the BASIC ROM. I am not going to recommend that everyone goes this far, but it does give the game a different feel, almost as if you have a new computer, with coloured, lower-case letters and more lines to the screen.

What I'm doing in effect is including a new Operating System into the game, burning up a bit of memory maybe, but it pays off. Let us see what it has to do. The main loop of an adventure is very simple, you start at a location, it is described, adding moveable objects as appropriate, and for an illustrated adventure it is picured.

You then input a command, with as much scope to edit it as possible. This is processed, usually as a verb and object. Three main response areas are serviced, either an objection is displayed, an action carried out at the location or the location is changed. The program then loops to display a new location or the changed state of the old one.

We will take the actions in this order and see how machine coding is applied. First the description of the location. There are two main word tables in the game, a verb table, which will be used later, and a table of everything else. Many of you will be familiar with the idea of comrpessing text by tokenising. BASIC uses it for programs and it helps with adventures as well. I have created a word table that is very like a BASIC program if sent directly to the screen, a mixture of bits of sense with embedded graphic characters. This not only compresses it but discourages cheaters as well.

The first words in the table are the location names, followed by the names of objects you can manipulate, then other words or letter groups, followed by full location descriptions and other response comments. The elements are separated by zeroes. The table serves the program many ways, the first words are data for the detokenising routine and serve the noun recognition routine, and it provides all the responses to be displayed.

As the display is in high resolution, an intermediate text space is needed for word processing with the text in ASCII form. The full display routine works as follows. The calling program provides the number of the text string to be displayed, and the table is scanned from the start until that number of zeroes are passed.

The buffer is cleared and characters are transferred to the buffer in turn. If a code exceeds 127 it is deemed a token and the detokenising subroutine is called. This stores the present table pointer, subtracts 128 from the code, finds the appropriate definition string and lays it in until a zero is found. The table pointer is restored and text from the main string continued. The subroutine may call itself if it finds a token within a token definition.

Since most location descriptions contain the location names, names of other locations accessible from it and names of objects, it makes sense to give these nouns tokens and define them earlier in the table where they can also serve other word handling routines.

When the text has been expanded into the buffer it is then converted into lower case, except the first letter and any subsequent letters two spaces beyond a full stop. This is easy using machine code with indexed addressing. The contents of the buffer, in standard ASCII rather than Dragon screen codes, is now treated for word wrap-round.

A register is pointed at the last character in the first line, and a counted backwards until a space is reached. All the subsequent text is moved up by this count and the end of the line filled with spaces. More lines are done until a zero is encountered.

The text is now ready to go on the screen. In this game the screen is divided so that the top half is for pictures, and apart from the input space at the bottom, the rest is for output text. The system counts the lines of the new material in the buffer, scrolls old text up by this amount, and inserts the new text, converting it using the hi-res character shape table. A marker is applied so the player sees where the new text starts.

The result is that best use is made of the text space, in most cases the player can see previous responses, and that, although the characters are a bit crude, the mixed cases give a very legible display. There is no need to preform the text to suit line length so descriptions can be composed freely.

A special character is used in the location descriptions which causes a diversion of the display routine during the expansion stage. This marks the beginning and end of references to moveable objects. For example you might want to say: YOU ARE IN THE KITCHEN. YOU SEE KNIVES. The table entry might expand to: YOU ARE IN THE KITCHEN. #YOU SEE#

When the first # is reached, the routine looks to see if there is anything in the kitchen and if not regards the first # as a zero and stops. Otherwise it substitutes a space and continues.

When it finds a # followed by a zero it lists the objects, each followed by a comma and space, except the last, which gets a full stop. To improve the sense for more than one object the word AND is needed before the last.

Best methods

With the description up the next stage is to draw a picture. One cannot deny that graphics are very greedy of memory, and in hi-res great cunning is needed to compress the data enough to stop it driving out the plot of the game. It depends a great deal which micro you have as to the methods that are best.

In the armoury are a few that are worth listing. "Run Length Encoding" means scanning the pixels and recording how many there are before a colour change, along with a code for the colour. Machine code is essential here, BASIC would take forever and a day. On a four colour display you could use one byte per element; the two most significant two bits for the colour, leaving the lower bits for a count up to 64 pixels. This coding is most

```
    3-- 6-- 9--12  21--24--27      41--42--43--44
    !           !   !   !           !           !
1-- 2-- 4--10  15--18  26--28--29  40--39  38  45  50
    !       !           !       !       !   !   !   !
    5-- 7  13  17      25--23  30--31  37--36  46--49
    !           !           !       !       !   !   !
    8--11--14--16--19--20--22  33--32--34--35  47--48
```

A maze map which the author has designed as a personal aid in writing an adventure

effective with horizontal strips, but, in the worst case, fine vertical detail, it uses more memory than the original.

Alternatively one can EXCLUSIVE OR a byte with the one above. If the result is not zero you store it. If it is zero you store a zero and go on do the next byte, counting rather than storing until a non-zero result comes. You then store the count, followed by the non-zero result, and loop.

This may seem a little complex to grasp, but it fares well with vertical repitition. You can also imitate BASIC's LINE, PLOT and FILL (PAINT) systems, or use special routines to generate the pictures. The latter may well be esential in mazes. Of course combinations may help, drawing over standard backgrounds downloaded from compressed data.

With the picture on the screen, the description displayed and the objects listed, it remains to input the player's command. This is typed in and the characters have to be echoed in the input space.

Verb search

The text buffer is used again here, the results of the keyboard scan are put into it and the whole contents redisplayed on every change. Machine code is fast enough to make this look instant. The cursor is indicated by a colour change and can backspace non-destructively for editing.

When the ENTER key is pressed, the command is processed. The game accepts single letter commands for direction, inventory and Look (in the sense of "repeat location description"). These are creamed off first, then a verb search is begun. Word searches start scanning the tables looking for a first letter match, a considerable time saving over BASIC.

When this tallies, an attempt is made to match the rest. As soon as this fails the first letter search continues. Success in matching is when either a zero is found in the table string, or a full stop is found in the buffer string. Thus input words can be abbreviated at the players risk, ie if he gives too few letters a match might be made on an earlier word in the table than he intended.

In the Verb Search a count is kept of the elements tested and this token is stored in workspace as the first word of the sentence. If no match is found an error message is given.

The "Noun" Search is repeated with all subsequent words, if the verb requires an object. This is slightly more flexible than the verb search. It searches the main word table and regards spaces as zeroes as it goes, and keeps no count. When a match is found it might be on the second word of a string. The table pointer is then run back to the start only counting zeroes to get the noun value. Thus a table might be:

KITCHEN
PANTRY
BEER CELLAR
DINING ROOM
BILLIARDS ROOM

The input might be:

GO CELLAR

The match will be on the fourth word, but on the return the pointer does not count the space between BEER and CELLAR and so gives the right string number. The player who expects to reach the BILLIARDS ROOM by typing "GO ROOM" has only himself to blame.

This style of entry is intended to alleviate the frustration for lazy typists like myself, and those whose logic dictates that "GO BEER" might rightfully lead to a merry drowning. However, there are penalties for laziness and it is part of the game to see what you can get away with. There is also the chance to use sentences of more than three words, such as "PUT DAGGER IN BOX".

This leads me to an analysis of the command once it has been tokenised. As with using BASIC, each verb leads to a very special action, and as such really needs a routine of its own to find out if the object and location are appropriate, then adjusting the data accordingly. Such routines often sprout branches that display one of the many disapproving comments in the main word table.

Two verb types form the core of most adventures, those to change your location, and those to move objects. The former include abbreviations N, E, S and W, and words such as GO, CLIMB, RIDE etc. All verbs involve "feasibility data", and in the case of GO and direction letters I have another table. We'll call it the LOCATION TABLE, but it is one of many types of MAPPING TABLE.

Each element starts with the string number for the location's description, and a byte to select the picture. Then follow codes for the possible exits, giving the locations reached, either as their number, or relatively, ie the number of elements to skip to reach the one in question. This scan, if the table is ordered correctly, allows more locations than the capacity of the code.

The verbs have to select the correct code in the element. In other words there has to be a way to identify which location is north, or which can be a valid object for GO. This can be by position in a fixed element length table, or by testing certain bits with the code.

There are mazes in the game I am writing, and I have condensed the data into a very compact table, which use relative addressing of neighbouring locations. The accompanying illustration shows a map of this type of maze and the numbering system used. No step in this maze involves location number changing by more than plus or minus three, yet there are 50 locations.

The keyboard scan I use for the Dragon gives a unique code for the arrow keys when shift is pressed, and players can steer themselves on the arrow keys using the shift, rather than move the cursor. Left and right turn you, and up arrow steps you forward.

Turning to verbs that manipulate objects, we need to think how to handle moveable object data. An object can only be in one place at a time so it is most compact to create a workspace table with one location element an object. We find out where an object is by reading the address formed by adding the object number to the table base address.

Assuming you don't go for more than 250 locations, a byte per element will suffice. The remaining values can stand for special locations viz:

Your are carrying it
It is in some container
You are wearing it
It is non-existent

Machine code searches can quickly scan for these codes and determine what and how much you are carrying, the contents of a box, what's in the kitchen and so on.

Graphics

Verb action is simple then, to GET something, check that you are in the location given for it in the table, count how many articles you are carrying and if not too many, put the CARRYING code into the table. The location description routine will then not find the object, and so it will not be "seen" in the description.

I'm probably preaching to the converted when I say that planning before coding is important, particularly in the verb action area. For instance as you narrow down the verbs, take those that work on a moveable object as a group, and look at the object for validity before beginning the individual verb routines. Then distribute the action to the routines with further object selection as required.

I haven't mentioned graphics animation, nor the value of machine code for real time adventures, where you don't want to penalise the player for time taken for BASIC to search strings. I think though you will see how machine code can enhance an adventure, giving that quickness and polish that make BASIC versions look old hat. □

A game with an unlimited life

A wargame that promises entertainment for months is how Tony Bridge describes Apocalypse in the review below

THE TROUBLE with most strategic-tactical wargames, compared with arcade games, is that once the solution is found, the game is usually consigned to a deep drawer. It has lost any surprise.

Red Shift have come up with a novel solution to this. Apocalypse is a game for 1-4 players — an interesting departure in itself, but solo players should be warned that solo play is, as they say, difficult. The game is produced in conjunction with Games Workshop. Available for the Spectrum 48K and BBC B what makes it particularly rewarding is that expansion kits are available which effectively give the game an unlimited life.

The master program comes in a stout box, roughly the size of two cassettes, which in my opinion is the best method of packaging nowadays: neat, endurable, good-looking and manageable. It costs £9.95.

In computer wargaming, as in board wargaming, bigger is often thought to be best, and Red Shift draw your attention to the fact that, with expansion modules, the whole outfit will give you over 400K of program and data. That should be enough for anybody.

The main, skeleton program is loaded first (during which you may choose to see the ads), after which one of the four scenarios contained in the master pack is loaded. You may elect to start wars in Europe, Britain, London or the Caribbean (I would have thought the Middle East would be more appropriate here). The idea is that, although the maps may differ, the game mechanics remain the same from map to map.

To start the game, players choose the millennium in which they wish to destroy the world: from 0AD (were nuclear weapons a problem then?) to 7999 AD. Then they decide on names for their empire, after which one of 10 symbols is given to each domain. Twenty-four empire centres are flashed on to the screen in quick succession, with each player attempting to grab one. If your reflexes are slow, you may find that you have got Tunis instead of Vienna. This carries on until all centres are occupied. I would have liked to have seen this come about as a result of strategy and territory-winning, rather than quickness of the hand. As it is, there seems to be little chance of coherent strategy at this stage.

Then the game proper begins. Each player deploys his armies, expanding during the course of the game from his chosen centres. In one corner of the screen is a box which shows a magnified, nine-square view of the area with information on each unit at the present location, underneath a cursor which the player can move around a screen.

Now old board wargamers (or old bored wargamers, or old BALD wargammers) are on familiar ground. The units look just like those little cardboard counters, with the strength of the unit displayed next to a little graphic symbol. The unit may be either navy, army or nuke.

During the deployment phase, new units may be built, depending on the revenue at the location (urban areas generating more than desert areas, and so on; the presence of nuclear weapons drastically reduces that area's earning capability. After all, who wants to live and work at ground zero).

After building and deployment comes moving and combat. All this is familiar ground — terrain affects movement, and the outcome of combat is decided by the players choosing one of several options. This is where the usual stumbling-block of multi-player computer games is met. At some point, everyone has to turn away while the current player makes his secret entry.

Combat in Apocalypse is a sort of scissors-and-paper game, comprising a secret input by attacker and defender of one from a list of numbers, followed by the matching of the options selected. If the defender has chosen the same option as the attacker, then the defender wins. If he underestimates the attack, the attacker wins, but may suffer some casualties. However, if the defender overestimates ▷



◁ then the attacker wins outright. The solo player may ask the computer to launch its own attack and defence systems, in which case, all you have to do is sit back and watch everything happen.

This is all pretty much like any other board game, but Apocalypse, as you will have realised, has a nuke capability. At any time during his move, a player may decide to launch a pre-emptive nuclear strike (the manual, reassuringly, warns agains the consequences) but doing so may well precipitate the final conflagration.

Complex rules

The manual is generally quite good though a little confusing in certain areas. It states that there are three basic sets of victory conditions then goes on to list just two. The first is the fast game, which consists of players fighting swift actions to occupy a certain number of (named) centres. No provision is made here for solo play. The second is the long game, which consists of an economic battle in which the players aim for a target revenue.

The rules are fairly complex, and so should appeal to refugees from the game board. Play ability, too, is good, and benefits from the colourful graphics: the manipulation of the various modules is a little awkward, and you may find yourself loading and reloading the programs a tedious chore from cassette.

However, Apocalypse and its attendant modules are extremely good value. The initial program will cost you, as mentioned, £9.95. There are, at the moment, three expansion modules available, each containing several new maps. These are £4.95 each, which is good value for the first volume since it contains USA, SE Asia, S Africa, the Artic Circle, Star Systems, the Nether Earth, a great selection. But subsequent volumes seem a little mean, offering, as they do, just two maps each, these cover the Fall of Rome, Napoleon's campaigns, the War in the Pacific, as well as 1984. With others to follow, no-one can complain about Apocalypse wearing out.

The maps are accessed by, first of all, loading the main program, and then the desired expansion kit. A sheet of instructions with the expansion kits gives victory conditions (which are different in each case) for each map and hints on how to play for the victory. The play mechanics, however, are the same for all the maps.

So, for instance, in the Galactic module, the player moves around a map of deep space, in which the empire centres are not earthly cities, like most of the others (with the exception of the Nether Earth scenario) but stars and planets. Learn the mechanics of the original tape, and the rest follows on. This is rather like the good old board games from Avalon Hill and SSI. Although the loading can be a little tricky, once play is started, everything is pretty straightforward.

To take a detailed look at just one of the scenarios, Decline and Fall, we can see that this is all about the dread Goths and Sassanids, against the Romans.

The map displays Europe in 286 AD, and the game is played through quarterly periods, and the play follows the same shape as that in the Master Module. There are certain details, however, in Decline and Fall, one being the emergence of disease. At the start of each quarter there is a random chance of plague occurring, and if this does indeed happen the effects will last for five periods.

Up to three plague areas may be effected at any one time, and troops within these areas will also be adversely affected. Combat, too, is slightly different, in that Roman Legions, if stronger than 10 divisions, may regroup after a wrong defence by their commander, and re-attack. The other scenarios have their own victory conditions, and their own anomalies.

Instructions

Though the original, Master Module is lavishly packaged, with a good manual, the expansion kits come in ordinary cassette boxes, with a sheet of photocopied instructions — not very inspiring.

All in all, however, what this adds up to is a game system which should keep anyone happy for many months. If you are fed up with saving the world from marauding alien hordes and want instead to start the final holocaust. □

Generation gap widens

Jason Orbaum looks at new horizons for adventures and offers hints on theme development

ADVENTURES HAVE been developing since the Crowther and Woods original mainframe game Adventures. The first micro computer adventures in the UK were for the ZX-81 and were usually written in Basic. These adventures had to stand up on the descriptive powers of their programmers and the fiendishness of the puzzles they set.

As can be expected, many were poor, even Psion's double cassette proved to be bad value for money. They just couldn't hold the attention of the average player long enough.

In those, now distant days, there were two types of game, arcade and adventure, and people were fans of either one type or the other. The same segregation can still be found, but trying to determine what is an arcade game and what is an adventure is becoming more and more difficult.

Lumbering monster

The game that began to merge the borders was 3D Monster Maze, by J K Greye (brings back memories doesn't it?). The idea was to escape from the maze, avoiding the hideous monster. It sounds like an adventure plot, but was played in three dimensions with superb use of the ZX-81's graphics in creating a terrifying, lumbering monster.

Then the same firm introduced Catacombs, a 'real-time graphic adventure', which was loved or hated by most people, but both its admirers and detractors consisted of arcade fans, and adventurers in equal number. This was the second generation of adventures: the graphic adventure with little text.

The idea caught on. Soon second generation adventures were found in arcades, with ther most popular being 'Tutankhamon, a fast action adventure set in a pyramid.

"But," the purists argued, "an adventure must have puzzles". The latest second generation adventures have enough puzzles to wrack even the most dedicated adventurer's brain. By way of an example allow me to use Raiders of the Lost Ark, a cartridge for the Atari 2600 system. Ask any owner of this game for help and he will load you with clues on how to reach the black market where you need the parachute, and how to get out of the marsh.

The third generation of adventure games features two games that stand out above other adventures. These are both available for the Spectrum and one of them has been translated across a range of machines. They are Valhalla and The Hobbit.

The third generation is the mixed text/graphic adventure. The Hobbit started it and was generally accepted as brilliant graphically. The game was heralded as one of the best adventures ever devised. This, in my opinion, is a vast exaggeration. The Hobbit is more than an average adventure. The independent characters don't really seem to do much except wait, move, tell you to hurry up, or sing about gold, and if Elrond says hello to me once more I shall scream. (Surely a flag to test how many times you have met him could have been inserted at little extra trouble.)

Good viewing

Valhalla suffers from the same faults as The Hobbit. It is brilliant to watch, but boring to play. Sorry Legend but that is the impression a number of people have, not just me.

The best third generation game in my opinion are in the Mysterious adventure series. Their strong point is that they started as text adventures and the plots

Overcoming writers' block

ONE THING many programmers complain about is lack of inspiration. Inspiration is essential to the adventure writer who needs a good scenario and plot. To help along any readers finding it difficult to develop their own scenarios I offer my own description of The Voyage.

Here begins The Voyage. We had been sailing against relentless winds. Every night when we dropped anchor the wind would try to pull us off course. The rations were low and murmurs of mutiny were heard in the cabins, but still I, as the captain, drove them on. I believe that I would be dead now if we hadn't stumbled accidently across the wreck.

It was on the morning of the tenth day of our voyage. The sea lay, like a great desert, on all sides of us. We stood out on the waves, a solitary craft on miles of ocean. I gave the order to pull up the anchor but it would not budge.

We sent down some divers to discover what was wrong and they came back talking of treasure and shipwrecks.

I dived that afternoon and found the wreck. Somehow air had been trapped in its upturned body so it was possible to surface for breath occasionally. I found what was holding the anchor down. It had jammed into the base of the ship and would need a team to get it out.

I surfaced, but to my horror the worst had happened. The crew had mutinied and gone off leaving me alone. They had cut the anchor rope and I saw it lying there, pushed into a never ending kaleidoscope of shapes by the waves.

Futuristic craft

Again I dived into the murky depths and back to the ship to explore it. Three hours later I had found several items which might have proved useful in the rooms that were not submerged. A musket, a gold chain, some knives, a tinder box and a chest that I could not open.

I decided to shoot the lock off the chest. Inside I found a map with a title that chilled me to my bones, Atlantis.

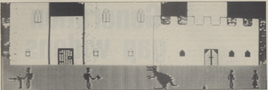
The whole ship shook and I couldn't believe my eyes as a futuristic craft came towards me. I felt a tractor beam pulling me in and then I passed out.

This game could be divided into three parts. In the first, the adventurer as captain of the ship must ration the food, and navigate a bath through the storms. If the player is lucky enough to drop anchor where the wreck is he can explore it in the second part.

This game could be played as a graphic adventure (second generation) in which the player must continually come up for air. When the player blows the lock off the chest the Atlantean ship takes him to the third adventure.

This adventure could take many forms, but must tie up the answers left by the former two adventures. These are: what caused the wreck of the ship in the second adventure? What has happened to the captain's original ship? Will the captain return home? What caused the appearance of the Atlantean craft when the player opened the chest?

The last adventure could be a struggle to escape Atlantis, during which the player finds out a great deal about the history of the place, or the player might wish to join the Atlantean community, for which he would have many tasks to fill. The possibilities are endless, once you have the inspiration.



have not changed. This means that they have graphics as an extra, whereas games such as Valhalla have graphics as one of the main features.

But what does the future hold for adventures? One thing is for sure, before the end of 1984 the multi-player game will have arrived, with or without Big Brother's help. This is the true fourth generation adventure. The adventure based on The Lord of the Rings may well need six players controlling Frodo, Gandalf, Aragorn, Fangorn, Denethor, and Theoden, with options for Saruman and Sauron to be played by either the computer or two more players.

The game, played on a disk-based network, allows communication between characters only when they are in the same location. The possibilities are endless. Imagine Frodo with the entire Shire community behind him offering the Ring to Sauron in return for Aragorn and Gandalf, and then not even keeping his side of the bargain.

And what of the fifth generation? Will it be arcade-style multi-player games played on a Vax based network? It could be that laser disks would make it possible to live an adventure in a much more realistic way than is possible at the moment.

These are already in use in the arcade games such as Astron belt, and a more recent release, Dragons Lair. This game is of interest to the adventurer as its plot is very like that of an adventure. The player controls Dirk the Daring in his struggles against dragons, bugs, black riders and the like to find a wizard and rescue a princess. Maybe this can be classed as a fifth generation adventure.

We could, of course, end up going full circle and returning to standard text adventures. This would be a good thing as long as the adventures were correctly constructed, but with the expected tide of Quill-like products rolling in from the silicon sea it will become increasingly easy to produce a poor adventure in an evening.

Having used the Quill I can see that all the preparation and planning of a good adventure will still have to be undertaken, but will software writers want to write a good adventure when they can make a fast buck from an average game? Time will sort the good software from the bad.□

YOUR ADVENTURES

Programs from readers this month include a text-only adventure from Alan Blackham in Stapleford **(this page)**. The program uses data statements. It should be suitable for any machine with the exceptions of a ZX81 and a Spectrum 16K. And from Jullian Marshall-James in Somerset comes On Safari, an adventure for the Vic 20, 16K. You'll find it on **page 45**. Some of the obstacles you'll meet are inhospitable terrains and unfriendly animals.

Send us your adventure listings — modules which readers can incorporate into their own games, short adventures and useful programming routines are all welcome. Please send us a printout and cassette along with a general description of the program and details of how it is constructed and can be used. If you want us to return your program, please enclose a stamped, addressed envelope. If you have any queries on the listings, write to the appropriate author, Your Adventures, Micro Adventurer, 12-13 Little Newport St, London WC2R 3LD

A mansion many enter but from which few successfully leave

From Alan Blackham in Stapleford, Notts

IN THIS text-only adventure, called Escape, it is the player's task to work through a labyrinth of rooms that make up a mansion that has fallen into disrepair.

On the way a player might encounter anything from monsters to trapdoors, before finding the key which opens the door leading to freedom.

As it uses data statements, this program should work on any computer except a ZX81.

It is too long for a Spectrum 16K.

Since it is a text adventure all that needs changing are CHR$(24) for clearing the screen. This is CLS on some computers or an inverse heart on the Commodore machines.

You are told where you are and you tell it what to do, such as GET CAT or N for go north.

The program will work on the Vic 20 with expansion, on the Spectrum 48K, the Commodore 64, the Dragon, the BBC and most other computers that use Basic.

```
10 REM          *******************
20 REM          *                 *
30 REM          * E S C A P E (C) *
40 REM          *                 *
50 REM          * BY ALAN BLACKHAM *
60 REM          *                 *
70 REM          * 19TH JAN 1984 (C)*
80 REM          *                 *
90 REM          *******************
95 GOSUB 9900
100 REM
110 REM ****** INITIALISE ******
120 REM
130 LET R=1
140 DIM T$(10),L(10),C$(10),R$(20)
150 DIM D1(20),D2(20),D3(20),D4(20)
160 FOR I=1 TO 20
170 READ R$(I)
180 NEXT I
190 FOR I=1 TO 20
200 READ D1(I),D2(I),D3(I),D4(I)
210 NEXT I
220 FOR I=1 TO 10
230 READ T$(I),C$(I),L(I)
240 NEXT I
250 REM ** DATA FOR ROOMS **
260 DATA BATH.ROOM,DINING.ROOM,KITCHEN,STORE.ROOM
270 DATA BED.ROOM,CELLER,DUST.BIN,COMPUTER.ROOM
280 DATA DARK.ROOM,LIVING.ROOM,SMALL.ROOM,LARGE.ROOM
290 DATA PLAY.ROOM,SWIMMING.POOL,GREEN.HOUSE,GARDEN
300 DATA GARAGE,SHED,DUSTY.ROOM,WENDY.HOUSE
310 REM ** DATA FOR DOORS **
320 DATA 0,2,0,0,0,3,0,1,0,4,0,2,0,0,7,3,0,6,0,0,0,0,9,5
330 DATA 4,0,10,0,0,9,13,0,6,0,0,8,7,11,0,0,0,15,12,10
340 DATA 11,0,19,14,0,14,0,0,0,12,17,13,0,0,0,11
350 DATA 0,17,0,0,14,18,0,16,0,19,0,17,12,20,0,18,0,0,0,19
360 REM ** DATA FOR THINGS **
370 DATA BOX,LOCKED,1,RAT,DEAD,3,TIN,FULL.OF.BAKED.BEANS,4
380 DATA GUN,NOT.LOADED,5,SAFE,LOCKED,10
390 DATA DOOR,LOCKED,20,SHOE,VERY.OLD.AND.DIRTY,14
400 DATA LADDER,BROCKEN.DOWN.THE.MIDDLE,12
410 DATA KNIFE,VERY.SHARP,16,MONSTER,VERY.BIG.!!!,19
420 REM
430 REM ****** START GAME ******
440 REM
450 PRINT CHR$(24)
460 PRINT"  E  S  C  A  P  E  !"
470 PRINT"--------------------------"
475 PRINT" BY A BLACKHAM."
480 PRINT:PRINT
490 PRINT"YOU ARE IN A ";R$(R)
500 PRINT
510 PRINT"EXITS ARE :";
```

```
520 IF D1(R)<>0 THEN PRINT'NORTH ';
530 IF D2(R)<>0 THEN PRINT'EAST ';
540 IF D3(R)<>0 THEN PRINT'SOUTH ';
550 IF D4(R)<>0 THEN PRINT'WEST ';
560 PRINT:PRINT
570 PRINT'YOU CAN SEE :'
580 FOR I=1 TO 10
590 IF L(I)=R THEN PRINT TAB(13);'A ';T$(I)
600 NEXT I
610 PRINT
620 PRINT'WHAT DO YOU WANT TO DO NOW ';
630 INPUT M$
640 IF M$='' THEN GOSUB 1490:GOTO 420
650 REM
660 REM ****** N , E , S OR W ******
670 REM
680 IF M$<>'N' AND M$<>'E' AND M$<>'S' AND M$<>'W' THEN 810
690 IF M$='N' AND D1(R)<>0 THEN R=D1(R):GOTO 420
700 IF M$='E' AND D2(R)<>0 THEN R=D2(R):GOTO 420
710 IF M$='S' AND D3(R)<>0 THEN R=D3(R):GOTO 420
720 IF M$='W' AND D4(R)<>0 THEN R=D4(R):GOTO 420
730 PRINT
740 PRINT'YOU CAN NOT GO : ';
750 IF M$='N' THEN PRINT'NORTH'
760 IF M$='E' THEN PRINT'EAST'
770 IF M$='S' THEN PRINT'SOUTH'
780 IF M$='W' THEN PRINT'WEST'
790 FOR X=1TO2000:NEXT X
800 GOTO 420
810 REM
820 REM ****** EXAMINE ! ******
830 REM
840 IF MID(M$,1,7)<>'EXAMINE' THEN 970
850 PRINT:PRINT
860 LET I$=MID(M$,9,20)
870 FOR I=1 TO 10
880 IF T$(I)=I$ THEN 930
890 NEXT I
900 PRINT'I CAN'T SEE A ';I$
910 FOR X=1TO2000:NEXT X
920 GOTO 420
930 IF R<>L(I) THEN 900
940 PRINT'  THE ';I$;' IS ';C$(I)
950 FOR X=1TO2000:NEXT X
960 GOTO 420
970 REM
980 REM ****** GET ! ******
990 REM
1000 IF MID(M$,1,3)<>'GET' THEN 1100
1010 LET I$=MID(M$,5,20)
1020 FOR I=1 TO 10
1030 IF I$=T$(I) AND R=L(I) THEN 1060
1040 NEXT I
1050 GOTO 900
1060 LET L(I)=100
1070 IF M$='GET KNIFE' THEN F1=1
1080 IF M$='GET KEY' THEN F2=1
1082 IF M$='GET TIN' THEN F3=1
1083 IF M$='GET WIRE' THEN F5=1
1085 IF M$='GET GUN' THEN 1800
1090 GOTO 420
1100 REM
1110 REM ****** LOOK ! ******
1120 REM
1130 IF M$<>'LOOK' THEN 1250
1140 PRINT CHR$(24)
1150 PRINT'YOU ARE CARRYING :'
1160 PRINT
1170 FOR I=1 TO 10
1180 IF L(I)<>100 THEN 1200
1190 PRINT TAB(10);'A ';T$(I)
1200 NEXT I
1210 PRINT:PRINT
1220 PRINT TAB(5);'PRESS 'RETURN' .'
1230 INPUT A$
1240 GOTO 420
1250 REM
1260 REM ****** DROP ! ******
1270 REM
1280 IF MID(M$,1,4)<>'DROP' THEN 1410
1290 LET I$=MID(M$,6,20)
1300 FOR I=1 TO 10
1310 IF I$=T$(I) AND L(I)=100 THEN 1370
1320 NEXT I
1330 PRINT:PRINT
1340 PRINT'  YOU HAVE NOT GOT A ';I$
1350 FOR X=1TO2000:NEXT X
```

A Vic 20 treasure safari

From Jullian Marshall-James in Somerset

ON SAFARI is an adventure for the Vic 20 16K in which the player must find King Solomon's treasure.

Wild animals inhabit the dark, damp, terrain which you are to explore. It contains a variety of locations, some set up with booby traps to hinder you in your search for the treasure. Be warned. Illusions may appear on your path to the treasure . . . all is not what it seems.

```
10 REM ON SAFARI
20 REM BY J.M.J 2/1/84
30 PRINT"YOU ARE AT YOUR BASE  CAMP."
35 PRINT"YOU ARE JUST LEAVING  TO FIND T
HE TREASURE."
40 PRINT"POSSIBLE EXITS:SOUTH, EAST,WEST
"
50 INPUT"WHAT SHALL I DO NOW?";WSN$
60 IF WSN$="GO SOUTH"THEN200
70 IF WSN$="GO EAST"THEN100
80 IF WSN$="GO WEST"THEN300
90 GOTO2715
100 PRINT"YOU ARE NOW AT A      RIVER CR
OSSING."
105 PRINT"ACROSS THE RIVER YOU  SEE A TR
EASURE CHEST."
108 PRINT"POSSIBLE EXITS:CROSS  RIVER,WE
ST."
110 INPUT"WHAT SHALL I DO NOW?";WSN$
120 IF WSN$="GO WEST"THEN30
130 IF WSN$="CROSS RIVER"THEN150
140 GOTO2725
150 PRINT"AS YOU CROSS THE"
155 PRINT"RIVER YOU ARE ATTACKEDBY PIRAN
HAS AND"
156 PRINT"KILLED."
160 GOTO1000
200 PRINT"YOU ARE NOW IN A"
205 PRINT"BEAUTIFUL CLEARING."
206 PRINT"YOU SEE:A GUN,KEY,AMMO"
207 PRINT"POSSIBLE EXITS:EAST,    NORTH,S
OUTH."
210 INPUT"WHAT SHALL I DO NOW?";WSN$
220 IF WSN$="TAKE GUN"THEN1400
230 IF WSN$="TAKE AMMO"THEN1500
240 IF WSN$="TAKE KEY"THEN1600
250 IF WSN$="GO NORTH"THEN30
260 IF WSN$="GO SOUTH"THEN450
270 IF WSN$="GO EAST"THEN400
280 GOTO2735
300 PRINT"YOU ARE NOW INSIDE A   DAMP CA
VE."
305 PRINT"YOU SEE:A BEAR AND CUB"
306 PRINT"POSSIBLE EXITS:SOUTH,  EAST."
310 INPUT"WHAT SHALL I DO NOW?";WSN$
320 IF WSN$="GO EAST"THEN30
330 IF WSN$="GO SOUTH"THEN520
340 GOTO2745
400 PRINT"AS YOU WALK ALONG YOU SINK IN
SOME QUICK    SAND AND DIE."
410 GOTO1000
450 PRINT"YOU ARE IN A EMPTY     CLEARIN
G."
455 PRINT"YOU SEE:A FEW ROCKS.    POSSIB
LE EXITS:NORTH,WEST."
460 INPUT"WHAT SHALL I DO NOW?";WSN$
470 IF WSN$="GO NORTH"THEN200
480 IF WSN$="GO WEST"THEN650
490 GOTO2755
520 PRINT"YOU ARE NOW AT YOUR      SUPPLY
 DEPOT."
525 PRINT"YOU SEE:YOUR SUPPLIES.  POSSIB
E EXITS:NORTH,  SOUTH."
530 INPUT"WHAT SHALL I DO NOW?";WSN$
540 IF WSN$="TAKE SUPPLIES"THEN1700
550 IF WSN$="GO NORTH"THEN300
560 IF WSN$="GO SOUTH"THEN650
570 GOTO2765
650 PRINT"YOU ARE NOW IN A SMALL CLEARIN
G.HERE YOU SET UP CAMP TILL MORNING."
655 PRINT"POSSIBLE EXITS:NORTH, SOUTH,EA
ST,WEST."
660 INPUT"WHAT SHALL I DO NOW?";WSN$
670 IF WSN$="GO NORTH"THEN520
680 IF WSN$="GO SOUTH"THEN730
690 IF WSN$="GO EAST"THEN450
700 IF WSN$="GO WEST"THEN800
```

```
1360 GOTO 420
1370 LET L(I)=R
1380 IF M$='DROP KNIFE' THEN F1=0
1390 IF M$='DROP KEY' THEN F2=0
1392 IF M$='DROP TIN' THEN F3=0
1393 IF M$='DROP WIRE' THEN F5=0
1400 GOTO 420
1410 REM
1420 REM ****** OTHERS ! ******
1430 REM
1440 IF M$()'OPEN RAT' AND M$()'CUT RAT' THEN 1490
1450 IF F1=0 THEN PRINT:PRINT' YOU HAVE NOT GOT A KNIFE
     !':GOTO 910
1460 PRINT:PRINT'THERE IS A KEY INSIDE ITS GUTS !'
1470 LET T$(2)='KEY':LET C$(2)='A DOOR KEY.'
1480 LET F2=1;GOTO 910
1490 IF M$()'OPEN BOX' AND M$()'UNLOCK BOX' THEN 1500
1491 IF F2=0 THEN PRINT:PRINT'YOU HAVEN'T GOT A KEY!'
     ;GOTO 910
1492 F4=1;PRINT'THERE ARE SOME NUMBERS WRITTEN IN THE BOX!'
1493 GOTO 910
1500 IF M$()'UNLOCK SAFE' AND M$()'OPEN SAFE' THEN 1510
1501 IF F4=0 THEN PRINT:PRINT'YOU DON'T KNOW THE
     COMBINATION!':GOTO 910
1502 PRINT:PRINT'THERE IS A PIECE OF WIRE IN THE SAFE!'
1503 T$(5)='WIRE';C$(5)='STIFF'
1504 GOTO 910
1510 IF M$()'FEED MONSTER' AND M$()'FEAD MONSTER' THEN 1520
1511 IF F3=0 THEN PRINT:PRINT'YOU HAVN'T ANYTHING TO FEED
     IT WITH!':GOTO 910
1512 F6=1;PRINT:PRINT'THE MONSTER VOMITS AND DIES
     PAINFULLY!'
1513 C$(10)='DEAD'
1514 GOTO 910
1520 IF M$()'PICK LOCK' AND M$()'OPEN DOOR' THEN 9000
1521 IF R()20 THEN PRINT:PRINT'I CAN'T SEE A DOOR TO
     OPEN!':GOTO 910
1522 IF F1=1 AND F2=1 AND F5=1 AND F6=1 THEN 1530
1523 PRINT:PRINT'I HAVEN'T ANYTHING TO OPEN IT WITH!'
1524 GOTO 910
1530 REM ** FINISHED ! **
1540 PRINT CHR$(24)
1545 FOR I=1 TO 100
1550 PRINT'WELL DONE!  ';
1555 NEXT I
1560 PRINT:PRINT:PRINT
1570 PRINT TAB(10);'YOU ESCAPED !'
1580 GOTO 10000
1800 REM ** DEAD **
1810 PRINT CHR$(24)
1815 FOR I=1 TO 30
1820 PRINT'BAD LUCK!  ';:NEXT I
1830 PRINT:PRINT
1840 PRINT TAB(10);'THE GUN WENT OFF IN YOUR HAND AND
     KILLED YOU!'
1850 FOR I=1 TO 5
1860 PRINT CHR$(7);:NEXT I
1870 GOTO 10000
9000 REM
9005 REM ****** DO NOT UNDERSTAND ! ******
9010 REM
9020 PRINT
9030 PRINT'I DO NOT KNOW HOW TO ';M$
9040 FOR X=1 TO 2000;NEXT X
9050 GOTO 420
9900 REM
9905 REM **** INSTRUCTIONS ****
9910 REM
9920 PRINT CHR$(24)
9925 PRINT TAB(20);'ESCAPE - BY A BLACKHAM.'
9930 PRINT TAB(19);'-----------------------'
9935PRINT:PRINT
9940 PRINT'ENTER COMMANDS SUCH AS GO , GET OR DROP .'
9945 PRINT
9950 PRINT'TO GO NORTH JUST ENTER 'N' OR EAST - 'E' ECT.'
9955 PRINT
9960 PRINT'ENTER 'LOOK' TO SEE WHAT YOU HAVE GOT .'
9965PRINT:PRINT
9970 PRINT'PRESS 'ENTER' TO BEGIN.'
9975 INPUT A$
9980 RETURN
10000 END
```

```
710 GOTO2275
730 PRINT"YOU ARE ON THE BANKS   OF A RI
VER UP STREAM  YOU SEE A WATERFALL."
735 PRINT"POSSIBLE EXITS:NORTH, EAST."
740 INPUT"WHAT SHALL I DO NOW?";WSN$
750 IF WSN$="GO NORTH"THEN650
760 IF WSN$="GO EAST"THEN2000
770 GOTO2785
800 PRINT"YOU ARE IN A LARGE       CLEARI
NG."
805 PRINT"POSSIBLE EXITS:EAST."
806 PRINT"SUDDENLY YOU ARE ATTAC-KED BY
LEOPARDS."
810 INPUT"WHAT SHALL I DO NOW?";WSN$
820 IF WSN$="SHOOT LEOPARDS"THEN1800
830 IF WSN$="GO EAST"THEN650
840 GOTO2795
1000 PRINT"HA,HA,YOU LOSE.NOW TRY AGAIN.
"
1100 GOTO30
1400 GU=1
1450 PRINT"YOU NOW HAVE THE GUN."
1460 GOTO210
1500 AM=1
1550 PRINT"YOU NOW HAVE THE AMMO."
1560 GOTO210
1600 KY=1
1650 PRINT"YOU NOW HAVE THE KEY."
1660 GOTO210
1700 SUP=1
1750 PRINT"YOU NOW HAVE SUPPLIES."
1760 GOTO530
1800 IF GU=1THEN1810
1805 PRINT"YOU HAVE NO GUN LEOPARDSEAT Y
OU."
1807 GOTO1000
1810 PRINT"LEOPARDS ARE DEAD."
1900 GOTO810
2000 PRINT"YOU ARE AT THE BOTTOM  OF A W
ATERFALL."
2005 PRINT"BEHIND THE WATERFALL   IS A C
AVE."
2006 PRINT"POSSIBLE EXITS:EAST,   CAVE."
2010 INPUT"WHAT SHALL I DO NOW?";WSN$
2020 IF WSN$="GO EAST"THEN2050
2030 IF WSN$="ENTER CAVE"THEN2500
2040 GOTO3100
2050 PRINT"YOU ARE ON THE BANKS   OF A R
IVER."
2055 PRINT"POSSIBLE EXITS:NORTH , EAST."
2056 PRINT"SUDDENLY YOU ARE ATTAC-KED BY
 PANTHERS."
2060 INPUT"WHAT SHALL I DO NOW?";WSN$
2070 IF WSN$="GO NORTH"THEN400
2080 IF WSN$="GO EAST"THEN400
2090 IF WSN$="SHOOT PANTHERS"THEN2200
2100 GOTO3200
2200 IF AM=1THEN2210
2205 PRINT"YOU HAVE NO AMMO YOU   ARE EA
TEN."
2207 GOTO1000
2210 GOTO2060
2500 PRINT"YOU ARE IN A CAVE YOU  SEE:A
LARGE CHEST."
2505 PRINT"POSSIBLE EXITS:WEST."
2510 INPUT"WHAT SHALL I DO NOW?";WSN$
2520 IF WSN$="GO WEST"THEN2560
2530 IF WSN$="OPEN CHEST"THEN2600
2540 IF WSN$="TAKE CHEST"THEN2700
2550 GOTO3300
2560 PRINT"AS YOU LEAVE THE CAVE  YOU SL
IP AND BREAK    YOUR NECK"
2570 GOTO1000
2600 PRINT"AS YOU OPEN THE CHEST  EVERTH
ING SPINS,AND"
2610 PRINT"YOU ARE BACKAT YOUR    CAMP,WI
TH THE TREASURE"
2620 PRINT"YOU HAVE FINISHED,WELL DONE!!
!!!
2630 PRINT"NOW TRY AGAIN"
2640 GOTO30
2700 PRINT"AS YOU TAKE THE CHEST  A GENI
E APPEARS,AND"
2705 PRINT"CAST A SPELL ON YOU."
2710 GOTO1000
2715 PRINT"CAN'T DO THAT"
2720 GOTO50
2725 PRINT"CAN'T DO THAT"
2730 GOTO110
2735 PRINT"CAN'T DO THAT"
2740 GOTO210
2745 PRINT"CAN'T DO THAT"
2750 GOTO310
2755 PRINT"CAN'T DO THAT"
2760 GOTO460
2765 PRINT"CAN'T DO THAT"
2770 GOTO530
2775 PRINT"CAN'T DO THAT"
2780 GOTO650
2785 PRINT"CAN'T DO THAT"
2790 GOTO740
2795 PRINT"CAN'T DO THAT"
3000 GOTO810
3100 PRINT"CAN'T DO THAT"
3150 GOTO2010
3200 PRINT"CAN'T DO THAT"
3250 GOTO2060
3300 PRINT"CAN'T DO THAT"
3350 GOTO2510
3400 STOP
```

Format 40/80 Club (BBC disc user group 5 Marsh Street, Bristol BS1 4AA. Adventure situation always included with monthly club disc. Sample issue £5.50. State drive format.

48K SPECTRUM. The Dragonstar Trilogy. Parts I to III. Classic Adventures, only £4.95 each! From Delta 4 Software, The Shieling, New Road, Swanmore, Hants SO3 2PE.

SAVE £££ ON SOFTWARE! FOR FREE LISTS SEND **SAE** TODAY TO: ACS., PO BOX AT 15, ANTRIM, COUNTY ANTRIM, BT41 1LT. (Please state computer model, eg: Spectrum, C64.)

TRADEWIND Graphic adventure for the **48K Spectrum.** Make a fortune sailing and trading in the coral islands. Risk your life among pirates and sharks! Avoid reefs, weather storms and outwit the local traders. £5 from WDSoftware(M), Hilltop, St Mary, Jersey, C.I.

CBM 64 PROGRAMS on disk: Witness, Deadline £20 each. On cassette Colossal Adventure, Adventure Quest £5 each. Telephone 0492 30250 after 6 p.m.

SPECTRUM OWNERS. Our software club has a complete range of cassettes from Digital Fantasia, R. Shepherd, Artic etc. Plus hundreds of pounds worth of other games and adventure programs. No rental charges. We also buy original tapes for half normal retail prices. Send list of tapes for sale or for more information send cheque or P.O for 50p to: Spectrumcette, Windermere, Cherrywood Road, Loughlinstown, County Dublin, Ireland.

AD INDEX

Here's my classified ad.

(Please write your copy in capital letters on the lines below.)

£0.20	£0.40	£0.60	£0.80
£1.00	£1.20	£1.40	£1.60
£1.80	£2.00	£2.20	£2.40
£2.60	£2.80	£3.00	£3.20
£3.40	£3.60	£3.80	£4.00
£4.20	£4.40	Please continue on a separate sheet of paper	

I make this .... words, at 20p per word so I enclose £ ....

Name..

Address..

....

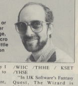
....

Telephone..

Please cut out and send this form to: Classified Department, *Micro Adventurer*, 12-13 Little Newport Street, London WC2R 3LD

ADVENTURE HELP

If you need advice or have some to offer write to Tony Bridge, Adventure Help, Micro Adventurer, 12-13 Little Newport St, London WC2R 3LD



A LETTER came recently from Irene Feeney, of Basildon in Essex: "I have just completed adventure 200, by Foilkade, following a plea for help in your column. Most letters I received were asking for help, realising that I had got further. However, it is a long and complicated adventure, and I would like to offer help to anyone who cares to write, enclosing an SAE.

Would it be possible, through your column, to thank Dougie Nisbet, John Price and Alan Frampton, who helped and swapped information."

Of course it would, Irene, and I'm glad that you got enough help to finish the adventure. This is what the Adventure Contract page should be about — adventurers, having solved all, or even part of a tough program, sharing their experiences with others who may be struggling along the same path, albeit some distance behind.

If you would like Irene to offer you help, write to her at: 16 Surrey Way, Laindon West, Basildon, Essex SS15 6PS.

Incidentally, spare a thought for this poor writer and let me know of any adventures that you have solved, on any machine, and your solutions. In this way I can give even more help to other readers.

And, to prove my point, here is a letter from Tony Ding, of Wickford, Essex. He is an Oric owner, and gives some valuable advice to other Oric adventurers. The program is Tevrog's Kingdom, from Cable Software. Says Tony: "You may find in your travels a large bag containing several items. This bag has the strange ability of being able to hold an enormous amount. Also, there is a large box, which has to be forced open with a crowbar, which has the same ability.

"Now on to Level 9's Colossal Adventure. To pacify the rather athletic bear: NFUE / NELD / OHCI / KMCS / HAAN / IDNW /WIIC /THHE / KSET /YHSE

"In IJK Software's Fantasy Quest, The Wizard is befriended by the wand, also, you must keep the rope until you're finished otherwise disaster will strike.

"I would be grateful if anyone can tell me where the pirate's chest is in Colossal Adventure since I'm going mad looking for it."

Well, Tony (Black Rider), help is, I hope, at hand. You'll find the chest once you have been robbed. When this happens, the pirate whips your treasure away to his chest. To find it, go to "the same maze". A diagonal move (eg, NE or SW), will bring you to the orange column, and you'll find the chest nearby.

Staying with Colossal Adventure, June Rowe, from Launceston, Cornwall, has offered a few cryptic clues for other adventurers who may be stuck. To open the clam: "It's no good pouring water on a shellfish so pour it where it's needed (in a pit). A long Trek then to refill — don't be selfish — you'll be rewarded, take my word for it. Now, before you start the climb, just pop next door to get some oil. This detour saves you travelling time and shows da way to git da poil.

"Please forgive the Brooklynese in the last line, couldn't think of a suitable rhyme for oil.

"I am stuck on Espionage Island, From Artic: where do I find beads to give to the native woman?"

Never mind about the Brooklynese, June, I lived in Brooklyn for five years and dat sounds great ta me! To get da beads for da goil — sorry, the beads, don't y'know for the young lady, you must feel around in the corner of the wrecked plane, then pull the string. Incidentally, I hope that you are over 18 — you'll need to be over a certain age to play certain adventures if some politicians have their way. Ever heard anything so ridiculous?

Happy adventuring. I hope the Grand Elf and I can give some more help next month. Now, to the Contacts.

ADVENTURE CONTACT

MICRO Spectrum 48K **Adventure** The Hobbit **Problem** How to find the boat; how to get the ring; how to get to laketown? **Name** Stephen Wood **Address** 81 Glenauon Rd, Birkenhead, Merseyside, L43 0RD.

MICRO Spectrum 48K **Adventure** The Hobbit **Problem** How do you avoid being killed by pale, bulbous eyes **Name** B Churchill **Address** 172 Mackenzie Rd, Beckenham, Kent.

MICRO Spectrum 48K **Adventure** The Hobbit **Problem** How do you get: Thrain's key; out of the goblin's dungeon and the dragon? **Name** David Kingdom **Address** 1 Tavern Close, Beetley, Dereham, Norfolk, NR20 4BN.

MICRO Spectrum 48K **Adventure** The Hobbit **Problem** Is it possible to open the portcullis? **Name** June Rowe **Address** 46 Hurdon Way, Launceston, Cornwall.

MICRO Commodore 64 **Adventure** Twin Kingdom Valley **Problem** With what do you kill the dragon in the desert king's castle? **Name** David O'Neill **Address** 15 Brokside Rd, Caton nr Lancaster, Lancs.

MICRO Vic 20 **Adventure** Pirate Cove **Problem** How do you open the chest and how do you get to treasure island? **Name** Andy Stubbs **Address** 95 Denvill Cres, Crossacres, Wythenshawe.

MICRO Vic 20 **Adventure** Curse of the Werewolf **Problem** Can't get past the zombie with the staff but the staff is rarely on my side of the river **Name** Paul King **Address** 122 Balnagasic Circle, Torry, Aberdeen.

MICRO BBC B **Adventure** Castle of Riddles **Problem** How to survive on the boat after the hang glider ride **Name** Graham Francis **Address** 32 Manningford Close, Winchester, Hants.

HAVE YOU BEEN staring at the screen for days, or given up in disgust, stuck in an adventure whose problems seem insurmountable? Adventure Contact may be the answer. This column is designed to put adventurers in touch with one another. When you're stumped a fellow adventurer may be able to help — and you may be able to solve other people's problems. If you are having difficulties with an adventure, fill in this coupon and send it to Adventure Contact, Micro Adventurer, 12/13 Little Newport St, London WC2R 3LD. We will publish Adventure Contact entries each month in this special column.

Micro ....

Adventure ....

Problem ....

....

....

Name ....

Address ....

....

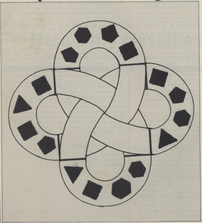
COMPETITION CORNER

Tony Roberts tests your skill — send your answers to Competition Corner, Micro Adventurer, 12-13 Little Newport St, London WC2R 3LD



This month Orion Data gives you the chance to keep abreast of new technology. Its Micro Command speech recognition unit allows you to input spoken commands into your Spectrum via a microphone (included in the package). And Orion Data is offering a complete unit free to each of the three competition solvers who devises the most appropriate tie-breaker.

Fortune offered in return for Tisch's precious D ring



IT SEEMS rather a shame to spoil your fun, but Tisch, the black dragon, has managed to materialise this jewelencrusted object right into the centre of the luxuriously appointed cave where she keeps you, a not too unwilling prisoner.

She has the first three Runic Rings and here is the fourth, if you can manage to remove it. It's simple really. Just remove two of the D-shaped loops, place them together and the D ring will be reformed. As usual you can keep the unwanted jewels.

One word of warning: if one of the unmatched D loops is removed before the D ring is formed, the entire object will simply case to exist.

Of course you do eventually manage to put the ring together successfully because you can see that the jewel stones on the two segments that form it follow a simple logical sequence. Which are they?

Your entry must arrive by the last working day in May. The winners and solution will be published in the July issue. You may enter only once. Entries will not be acknowledged and we cannot enter into correspondence on the result.

The winner of the March competition was Steven Brooks, of High Wycombe, Bucks.

The solution to the maze was the third block from the left on the bottom row. The logic is no group of three adjacent boxes shall be all marked white or black; square or triangle, A or B.

As a tie-breaker complete the following sentence in 30 words or less: "I want to input speech using Orion Data's Micro Command because . . ."

TIRED of endless battles with orcs?
CONFUSED by interminable corridors and limitless caves?
BORED by all the same old adventures?

TRY

The CRICKLEWOOD INCIDENT

and

Wings of War





two new and different adventures from Salamander Software.

In WINGS OF WAR you play Lt. Roger Wilcoe, reluctant hero. The time is November 1942 – you must parachute into occupied France, disguised as a German Officer, infiltrate a secret arms laboratory hidden in a chateau, steal the plans and the prototype of a new bomb, and make good your escape back to Blighty.

ROGER WILCOE
Star of Wings of War.

In THE CRICKLEWOOD INCIDENT, you, Arnold Q. Volestrangler the 14th heir to the Volestrangler fortune, find yourself bored out of your mind and looking for something to do until the laundrette opens. There's nothing for it but to leave your padded cell and search the wilds of London, Moscow, Hanoi and Wigan for the elusive Holy Grail. NOT for sheep of a sensitive disposition.

MAVIS NASTY
Not the star of the Cricklewood Incident.

Tape duplication by DATACLONE.

Stimulate yourself for only £7.95 each.

Please add 50p p&p to all orders. Send A5 SAE for full catalogue.

Cheques or postal orders payable to:

Salamander SOFTWARE

17 Norfolk Road, Brighton, BN1 3AA Telephone: Brighton (0273) 771942

Look for these other new releases from Salamander Software

DRAGON – DRS (sophisticated database) £14.95
TURTLE GRAPHICS (Educational & fun) £9.95

BBC – EAGLE (original arcade action) £7.95
BBC/ELECTRON – TURBO COMPILER (BASIC compiler) £9.95

Now available are versions of our best selling 737 Flight Simulator for BBC disk and ACORN Electron.


A SENSATIONAL ADVANCE IN COMPUTER GAME TECHNOLOGY
MICRO COMMAND
YOU SPEAK— THE COMPUTER OBEYS
Easily programmed voice control
Sensitive to individual voice prints
Instant response on screen
Your attention can stay on the screen – not on the controls
FOR USE WITH sinclair ZX Spectrum HOME COMPUTER UNITS
MAJOR PRICE BREAKTHROUGH ONLY £49.95 R.R.P.
Further Distributor, Dealer and Export Enquiries Welcome
ORION DATA
3 Cavendish Street, Brighton, East Sussex
BN2 1RN
Tel: (0273) 672994